P.J. Narayanan
Shree K. Nayar
Heung-Yeung Shum (Eds.)

LNCS 3852

Computer Vision – ACCV 2006

7th Asian Conference on Computer Vision
Hyderabad, India, January 2006
Proceedings, Part II

2 Part II

7TH ACCV 2006

Springer

P.J. Narayanan   Shree K. Nayar
Heung-Yeung Shum (Eds.)

Computer Vision – ACCV 2006

7th Asian Conference on Computer Vision
Hyderabad, India, January 13-16, 2006
Proceedings, Part II

Springer

Volume Editors

P.J. Narayanan
Centre for Visual Information Technology
International Institute of Information Technology
Gachibowli, Hyderabad 500032, India
E-mail: pjn@iiit.ac.in

Shree K. Nayar
Columbia University, Department of Computer Science
530 West 120th Street, New York, NY 10027, USA
E-mail: nayar@cs.columbia.edu

Heung-Yeung Shum
Microsoft Research Asia
49 Zhichun Road, Beijing 100080, China
E-mail: hshum@microsoft.com

Library of Congress Control Number: 2005938106

CR Subject Classification (1998): I.4, I.5, I.2.10, I.2.6, I.3.5, F.2.2

ISSN 0302-9743
ISBN-10 3-540-31244-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-31244-4 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11612704 06/3142 5 4 3 2 1 0

Preface

Welcome to the 7th Asian Conference on Computer Vision. It gives us great pleasure to bring forth its proceedings. ACCV has been making its rounds through the Asian landscape and came to India this year. We are proud of the technical program we have put together and we hope you enjoy it.

Interest in computer vision is increasing and ACCV 2006 attracted about 500 submission. The evaluation team consisted of 27 experts serving as Area Chairs and about 270 reviewers in all. The whole process was conducted electronically in a double-blind manner, a first for ACCV. Each paper was assigned to an Area Chair who found three competent reviewers for it. We were able to contain the maximum load on the reviewers to nine and the average load to less than six. The review form had space for qualitative and quantitative evaluation of the paper on nine aspects. The submitted reviews underwent an elaborate process. First, they were seen by the Area Chair, who resolved divergences of opinion among reviewers, if any. The Area Chair then wrote qualitative comments and a quantitative score along with his/her initial recommendation on the paper. These were looked at by Program Co-chairs and compiled into a probables list. The Area Chairs and Program Co-chairs met in Beijing during ICCV to discuss this list and arrived at the final list of 64 oral papers and 128 posters. Naturally, many deserving papers could not be accommodated.

Katsushi Ikeuchi has been unflinching in his support of ACCV as a whole and ACCV 2006 in particular. His help was critical at many stages. We must thank the Area Chairs and the reviewers for their time and effort towards the conference. From IIIT Hyderabad, C.V. Jawahar and Anoop M. Namboodiri contributed in many ways with the program. The enthusiastic team of students from the Centre for Visual Information Technology (CVIT) was behind it fully. Karteek Alahari, Kiran Babu Varanasi, Sumeet Gupta, Sukesh Kumar, and Satyanarayana made all the logistics of the CFP, paper submission, review process, and preparation of the proceedings really possible. The International Institute of Information Technology was fully behind the conference as a team and deserves our deep gratitude. Finally – but most importantly – we wish to thank the authors who showed great enthusiasm for ACCV.

ACCV has been gaining in stature as a platform to showcase the best of computer vision research over the years. We hope the 2006 edition has brought it forward at least a little. Computer vision continues to be an exciting area and conferences like these provide the much needed light to many who will embark on a journey down its path.

P J Narayanan
Shree Nayar
Harry Shum
(Program Chairs)

Conference Committees

General Chairs
Narendra Ahuja
University of Illinois & IIIT Hyderabad
Takeo Kanade
Carnegie Mellon University
Tieniu Tan
Chinese Academy of Sciences

Program Chairs
P.J. Narayanan
IIIT, Hyderabad
Shree Nayar
Columbia University
Harry Shum
Microsoft Research Asia

Organizing Chairs
C.V. Jawahar
IIIT, Hyderabad
Santanu Chaudhury
IIT, Delhi

Advisory Committee
Masahiko Yachida, *Osaka University*
Eam Khwang Teoh, *Nanyang Technological University*
Roland Chin, *Hong Kong University of Science and Technology*
Wen-Hsiang Tsai, *Chiao Tang University*
David Suter, *Monash University*
Sang-Uk Lee, *Seoul National University*
Katsushi Ikeuchi, *Tokyo University*
B. L. Deekshatulu, *University of Hyderabad*
D. Dutta Majumdar, *Indian Statistical Institute*
B. N. Chatterjee, *Indian Institute of Technology, Kharagpur*

Area Chairs

Yaron Caspi	*Hebrew University*
Tat Jen Cham	*Nanyang Technological University*
Bhabatosh Chanda	*Indian Statistical Institute*
Subhasis Chaudhuri	*Indian Institute of Technology, Mumbai*
Yi-ping Hung	*National Taiwan University*
Prem Kalra	*Indian Institute of Technology, Delhi*
Chandra Kambhamettu	*University of Delaware*
Mohan Kankanahalli	*National University of Singapore*
In So Kweon	*Korean Advanced Institute of Science and Technology*
Sang Wook Lee	*Sogang University*
Ravikanth Malladi	*GE John Welch Technology Centre*
Hiroshi Murase	*Nagoya University*
Tomas Pajdla	*Czech Technical University*
Long Quan	*Hong Kong University of Science and Technology*
A.N. Rajagopalan	*Indian Institute of Technology, Madras*
Mubarak Shah	*University of Central Florida*
Takeshi Shakunaga	*Okayama University*
David Suter	*Monash University*
Tanveer Syeda-Mahmood	*IBM Almaden Research Center*
Chi-Keung Tang	*Hong Kong University of Science and Technology*
Xiaoou Tang	*Microsoft Research Asia*
Rin-ichiro Taniguchi	*Kyushu University*
Baba Vemuri	*University of Florida*
Yaser Yacoob	*University of Maryland*
Naokazu Yokoya	*Nara Institute of Science and Technology*
Changshui Zhang	*Tsinghua University*
Zhengyou Zhang	*Microsoft Research, Redmond*

Reviewers

Neeharika Adabala
Manoj Aggarwal
Amir Akbarzadeh
Yusuf Akgul
Kenichi Arakawa
Greg Arnold
Naoki Asada
Mark Ashdown
Tarkan Aydin
Noboru Babaguchi
Simon Baker
Hynek Bakstein
Alok Bandekar
Subhashis Banerjee
Musodiq Bello
Kiran Bhat
Rahul Bhotika
Prabir Kumar Biswas
Michael Brown
Sema Candemir
Lekha Chaisorn
Kap Luk Chan
Michael Chan
Sharat Chandran
Peng Chang
Parag Chaudhuri
Datong Chen
Chu-Song Chen
Xilin Chen
Yong-Sheng Chen
James Cheong
Tat-Jun Chin
Ondrej Chum
Antonio Criminisi
Shengyang Dai
Kristin Dana
James Davis
Amadou Diallo
Gianfranco Doretto
Lingyu Duan
Sumantra Dutta Roy
Ryan Eckbo
Alexei Efros
Hazim Kemal Ekenel
Sabu Emmanuel
Chris Engels
Mark Everingham
Zhimin Fan
Jan-Michael Frahm
Kazuhiro Fukui
Hui Gao
Theo Gevers
Christopher Geyer
Joshua Gluckman
Dmitry Goldgof
Girish Gopalakrishnan
Ralph Gross
Yanlin Guo
Keiji Gyohten
Mei Han
Wang Hanzi
Manabu Hashimoto
Jean-Yves Hervé
Shinsaku Hiura
Jeffrey Ho
Ki-Sang Hong
Anthony Hoogs
Osamu Hori
Kazuhiro Hotta
Changbo Hu
Gang Hua
Rui Huang
Szu-Hao Huang
Daniel Huber
Sei Ikeda
Ali Iskurt
C.V. Jawahar
Jiaya Jia
Seon Joo Kim
Ioannis Kakadiaris
Atul Kanaujia
Masayuki Kanbara
Moon Gi Kang
Sing Bing Kang
Mark Keck
Zia Khan
Ron Kimmel
Koichi Kise
Dan Kong
Ravi Kothari
Ryo Kurazume
Uday Kurkure
James Kwok
Shang-Hong Lai
Arvind Lakshmikumar
Shihong LAO
Kyoung Mu Lee
Wee Kheng Leow
Maylor Leung
Thomas Leung
Dahua Li
Liyuan Li
Min Li
Lin Liang
Chia-Te Liao

Jenn-Jier James Lien
Joo-Hwee Lim
Stephen Lin
Che-Bin Liu
Zhiheng Liu
Qingshan Liu
Tyng-Luh Liu
Xiaoming Liu
Zicheng Liu
Yogish Mallya
Jose Marroquin
Daniel Martinec
Bogdan Matei
Yasuyuki Matsushita
Scott McCloskey
Paulo Mendonca
Shabbir Merchant
Branislav Micusik
Karol Mikula
James Miller
Anurag Mittal
Daisuke Miyazaki
Kooksang Moon
Yasuhiro Mukaigawa
Dipti Prasad Mukherjee
Jayanta Mukhopadhyay
Kartik Chandra Muktinutalapati
Rakesh Mullick
Christopher Nafis
Anoop Namboodiri
Srinivasa Narasimhan
Ko Nishino
David Nister
Naoko Nitta
Takahiro Okabe
Shinichiro Omachi
Sean O'Maley
Taragay Oskiper
Jiazhi Ou
Dirk Padfield
Kannappan Palaniappan
Vladimir Pavlovic
Shmuel Peleg
A.G. Amitha Perera
Michael Phelps
Carlos Phillips
Marc Pollefeys
Daniel Pooley
Arun Pujari
Kokku Raghu
Deepu Rajan
Subrata Rakshit
Srikumar Ramalingam
Ravi Ramamoorthi
Visvanathan Ramesh
Anand Rangarajan
Sohan Ranjan
Cen Rao
Christopher Rasmussen
Alex Rav-Acha
Sai Ravela
Jens Rittscher
James Ross
Amit Roy-Chowdhury
Hideo Saito
Subhajit Sanyal
Alessandro Sarti
Imari Sato
Tetsu Sato
Tomokazu Sato
Yoichi Sato
Peter Savadjiev
Konrad Schindler
Andrew Senior
Erdogan Sevilgen
Shiguang Shan
Ying Shan
Vinay Sharma
Zhang Sheng
Sheng-Wen Shih
Ikuko Shimizu Okatani
K.S. Shriram
Kaleem Siddiqi
Terence Sim
Sudipta Sinha
Jayanthi Sivaswamy
Thitiwan Srinark
S.H. Srinivasan
Christopher Stauffer
Jesse Stewart
Henrik Stewenius
Svetlana Stolpner
Peter Sturm
Akihiro Sugimoto
Rahul Sukthankar
Qibin Sun
Srikanth Suryanarayananan
Bharath Kumar SV
Rahul Swaminathan
Gokul Swamy
Kar-Han Tan
Ming Tang
Hai Tao

SriRam Thirthala
Ying-Li Tian
Prithi Tissainayagam
George Toderici
Shoji Tominaga
Wai Shun Dickson Tong
Philip Torr
Lorenzo Torresani
Emin Turanalp
Ambrish Tyagi
Seiichi Uchida
Norimichi Ukita
Anton van den Hengel
Rajashekar Venkatachalam
Svetha Venkatesh
Ulas Vural
Toshikazu Wada
Meng Wan
Huan Wang
Liang Wang
Shu-Fan Wang
Chieh-Chih (Bob) Wang
Zhizhou Wang
Tomas Werner
Frederick Wheeler
Kwan-Yee Kenneth Wong
Woontack Woo
Wen Wu
Yihong Wu
Ying Wu
Jing Xiao
Jiangjian Xiao
Wei Xu
Yasushi Yagi
Shuntaro Yamazaki
Kazumasa Yamazawa
Shuicheng Yan
Hua Yang
Ming Yang
Changjiang Yang
Jie Yang
Ming-Hsuan Yang
Ruigang Yang
Qingxiong Yang
Jieping Ye
Dit-Yan Yeung
Ting Yu
Xinguo Yu
Jingyi Yu
Ali Zandifar
Xiang Zhang
Hongming Zhang
Li Zhang
Tao Zhao
Wenyi Zhao
Jiang Yu Zheng
Wei Zhou
Yongwei Zhu
Andrew Zisserman
Larry Zitnick

Table of Contents – Part II

Detection and Applications

Statistics and Kernels

Segmentation

Geometry and Statistics

Signal Processing

Poster Session 3

Video Processing

Table of Contents – Part I

Camera Calibration

Stereo and Pose

Texture

Poster Session 1

Face Recognition

Variational Methods

Tracking

Geometry and Calibration

Lighting and Focus

Poster Session 2

Infinite Homography Estimation Using Two Arbitrary Planar Rectangles

Jun-Sik Kim and In So Kweon

Dept. of EECS, KAIST,
373-1 Kusong-dong, Yuseong-Gu, Daejeon, Korea
jskim@rcv.kaist.ac.kr, iskweon@kaist.ac.kr
http://rcv.kaist.ac.kr

Abstract. In this paper, we propose a new method to estimate an infinite homography between two views containing two arbitrary planar rectangles. The proposed method does not require metric measurements, such as rectangle lengths or aspect ratios of the rectangles. We introduce the concept of semi-metric cameras and show that the semi-metric cameras derived from different views that see an identical 3D rectangle, can be regarded purely translating cameras whose pixel is zero-skewed. New parameterization for infinite homography is developed based upon the semi-metric space, and this parameterization is used to propose a new algorithm to estimate infinite homography. As a direct application, we apply our algorithm to autocalibration for a scene only with a few feature points on each rectangles.

1 Introduction

In the real world, there are many objects with two-dimensional planes and rectangular shapes, especially in outdoor urban environment. Cameras generally use planar CCD or CMOS type sensors. Therefore, the imaging process of planar objects can be described as a 2D to 2D transformation [1]. Furthermore, in the case of multiple views, the transformation between imaged planes can be also considered 2D to 2D, and is called *plane induced homography.* Plane induced homography offers a useful tool to describe scenes with planar objects from two or more views, as shown in plane + parallax approaches [1, 2, 3, 4, 5].

Among the plane induced homographies, a particularly important one is an *infinite homography.* The infinite homography is the homography induced by the plane at infinity with some important properties. First, it maps features on the plane at infinity of one view, such as vanishing points, vanishing lines and images of absolute conic, to another views. Second, it can be used to find affine and metric reconstructions from projective ones. This means we can calibrate a camera from image sequences using the infinite homographies. Additionally, we can reduce the search region for stereo matching through mapping with the infinite homography. Detailed explanations about these issues can be found in [1]. Note that the infinite homography between two views depends only on the rotation between the cameras capturing the views and the intrinsic parameters of them.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 1–10, 2006.

Three methods are commonly used to estimate the infinite homography between two views. The first method uses camera motion constraints. If we use a purely rotating camera to capture the images, the homography induced by any plane on the image is the infinite homography. Although this method is easy to apply, it requires the use of rotating cameras. The second method uses strong scene constraints. If we have three vanishing points in each view with a fundamental matrix, the infinite homography can be estimated. Similarly it can be calculated from corresponding vanishing lines and vanishing points with the fundamental matrix. This requires the identification of three vanishing points and vanishing lines, however it may be difficult to find the features in infinity. The third method is a stratified approach. Once we find an affine reconstruction and projectively transformed plane at infinity, we can find the infinite homography from the projective projection matrix and the plane at infinity. The most difficult part of this approach is to build an affine reconstruction from the projective one. It requires some constraints of the scene and the camera, or the modulus constraints [6] for a static camera.

In this paper, we propose a new method to linearly estimate the infinite homography from images containing two arbitrary rectangles. The term "arbitrary" implies that we do not have information regarding the lengths, the aspect ratios, and the relative poses of the two rectangles. This method uses information about the parallelism and orthogonality, however this method does not require finding the vanishing points or the vanishing lines explicitly, which can be difficult for some rectangles. Furthermore, estimating epipolar geometry is also not required to estimate the infinite homography. Only tracking two rectangles between two views is needed.

In Sect. 2, we introduce the concept of semi-metric cameras and discuss some of their properties, such as image of absolute conic and special form of camera matrix. Sect. 3 discusses ways to parameterize the infinite homography using semi-metric cameras and to estimate the infinite homography using the proposed parameterization with two imaged rectangles. In Sect. 4, we show an important application of the infinite homography - the autocalibration of cameras - using the proposed algorithm. We conclude this paper in Sect. 5.

2 Semi-metric Cameras

We have introduced the concept of semi-metric space, defined as the sub-space of affine space [7]. In semi-metric space, orthogonal features are preserved, however the aspect ratio between two orthogonal axes is not preserved.

Assuming that there is a rectangle with an unknown aspect ratio in 3D space and a view capturing the rectangle in a general position, we can find a homography to make the projectively distorted rectangle to align the orthogonal axis of the rectangle. The warped image is called as semi-metric image. To make semi-metric images, two methods are used [7].

The first method uses vanishing points whose directions are orthogonal to each other. Warping the vanishing points to infinite points makes a semi-metric image with warping matrix defined as

$$\mathsf{H}_{sm} = \left[\mathbf{v}_1 \; \mathbf{v}_2 \; \mathbf{x}_c\right]^{-1}$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ are vanishing points orthogonal to each other and $\mathbf{x}_c$ is an arbitrary point, as shown in Fig. 1.

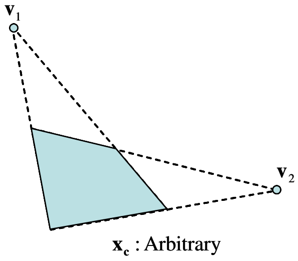


Fig. 1. Elements of semi-metric transformation matrix from vanishing points

Warping from the projected rectangle to a standard predefined rectangle with a known aspect ratio is sufficient to warp to a semi-metric image. Fig. 2 shows the concept of the warping method using a standard rectangle. Note that an aspect ratio of the warped rectangle can be set arbitrarily. For example, in Fig. 2, the aspect ratio is set to one.

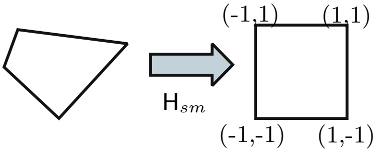


Fig. 2. Semi-metric warping using a standard rectangle

With a semi-metric image, the following theorem can be proven [7].

Theorem 1. *In semi-metric space, the ICDCP is given as* $\mathrm{diag}\left(R_m^2, R_{sm}^2, 0\right)$ *where* R_m *is the aspect ratio of the model rectangle, and* R_{sm} *is the aspect ratio of a semi-metric warped rectangle.*

Because the ICDCP in semi-metric space is expressed as $\mathrm{diag}(R_m^2, R_{sm}^2, 0)$, the imaged circular points (ICP) that is its dual feature, are simply expressed as

$$\mathbf{I_{sm}} = \begin{bmatrix} R_m \\ iR_{sm} \\ 0 \end{bmatrix}, \mathbf{J_{sm}} = \begin{bmatrix} R_m \\ -iR_{sm} \\ 0 \end{bmatrix}.$$

Furthermore, we can assume that there is a physical camera to make the semi-metric image. This camera is referred to as a *semi-metric camera.* To find some properties of semi-metric cameras, image of absolute conic (IAC) of semi-metric cameras is studied.

Assuming that there are three vanishing points $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ whose directions are orthogonal in 3D, then the IAC would be expressed as [8]

$$\boldsymbol{\omega} = \alpha^2 \mathbf{l}_1 \mathbf{l}_1^\top + \beta^2 \mathbf{l}_2 \mathbf{l}_2^\top + \gamma^2 \mathbf{l}_3 \mathbf{l}_3^\top$$

where α, β and γ are proper scale factors and $\mathbf{l}_1$, $\mathbf{l}_2$, and $\mathbf{l}_3$ are vanishing lines given as

$$\mathbf{l}_1 = \mathbf{v}_1 \times \mathbf{v}_2, \mathbf{l}_2 = \mathbf{v}_2 \times \mathbf{v}_3, \mathbf{l}_3 = \mathbf{v}_3 \times \mathbf{v}_1.$$

In semi-metric space, the vanishing points $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ can be set as

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix} a \\ b \\ c \end{bmatrix},$$

which gives us the IAC in semi-metric space $\boldsymbol{\omega}_{sm}$ as

$$\boldsymbol{\omega}_{sm} = \begin{bmatrix} \beta^2 c^2 & 0 & -\beta ac \\ 0 & \gamma^2 c^2 & -\gamma bc \\ -\beta ac & -\gamma bc & \alpha^2 + \beta^2 a^2 + \gamma^2 b^2 \end{bmatrix}. \tag{1}$$

Because the ICPs are on the IAC,

$$\mathbf{I}_{sm}^\top \boldsymbol{\omega}_{sm} \mathbf{I}_{sm} = 0, \mathbf{J}_{sm}^\top \boldsymbol{\omega}_{sm} \mathbf{J}_{sm} = 0,$$

and we can find the relation that

$$\frac{{R_m}^2}{{R_{sm}}^2} = \frac{\gamma^2}{\beta^2}.$$

This means that the ratio of β and γ is equal to that of R_{sm} and R_m. By decomposing (1), the camera matrix in semi-metric space is given as

$$\mathsf{K}_{sm} = \begin{bmatrix} 1/R_{sm} & 0 & a \\ & 1/R_m & b \\ & & c \end{bmatrix} \tag{2}$$

up to scale.

As a consequence, the camera matrix K_{sm} represents a camera whose skew is zero, and its pixel aspect ratio is equal to a ratio between the aspect ratio of the reference rectangle R_m and the corresponding semi-metric aspect ratio R_{sm}. The principal point of the camera is expressed with the scaled third vanishing point $\mathbf{v}_3$ and the scale plays the role of a focal length. In other words, the semi-metric camera matrix is determined with scene information and a camera pose.

Naturally, the relation between IAC $\boldsymbol{\omega}$ in projective space and IAC $\boldsymbol{\omega}_{sm}$ in semi-metric space is obtained from basic conic transformation as

$$\mathsf{H}_{sm}^{-\top} \omega \mathsf{H}_{sm}^{-1} = \omega_{sm} \tag{3}$$

where H_{sm} is a plane homography from projective space to semi-metric space.

3 Estimation of Infinite Homography

In this section, we derive a parameterization of an infinite homography in terms of semi-metric warping matrices. Using the parameterization, it is possible to estimate an infinite homography linearly from images of two arbitrary rectangles.

3.1 Parameterization of Infinite Homography

Assuming that there are two views containing a projected unknown rectangle, then each semi-metric camera matrix K_{sm1}, and K_{sm2} would be expressed as

$$\mathsf{K}_{sm1} = \begin{bmatrix} 1/R_{sm} & 0 & s_1 m_1 \\ & 1/R_m & s_1 m_2 \\ & & s_1 m_3 \end{bmatrix}, \mathsf{K}_{sm2} = \begin{bmatrix} 1/R_{sm} & 0 & s_2 n_1 \\ & 1/R_m & s_2 n_2 \\ & & s_2 n_3 \end{bmatrix}$$

using (2). Note that we can set the value of R_{sm} to 1, as explained in Sect. 2. Furthermore, since the plane is identical, R_m is the same in both K_{sm1} and K_{sm2}.

A semi-metric image is generated by simple image warping. We can find the projection matrix of semi-metric camera directly as

$$\begin{aligned} \mathsf{P}_{sm} &= \mathsf{H}_{sm}\mathsf{K}\left[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3\ \mathbf{t}\right] \\ &= \left[\mathsf{K}_{sm}\ \mathbf{e}_3\right] \\ &= \mathsf{K}_{sm}\left[\mathtt{I}_{3\times 3}\ \mathsf{K}_{sm}^{-1}\mathbf{e}_3\right] \end{aligned} \tag{4}$$

where $\mathbf{e}_3 = \left[0\ 0\ 1\right]^\top$. Note that all semi-metric cameras derived from an identical 3D rectangle are under pure translating motion. An infinite homography between two semi-metric cameras, K_{sm1} and K_{sm2} can be simply given as

$$\mathsf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & t_z \end{bmatrix},$$

because an infinite homography is generally given as [1]

$$\mathsf{T} = \mathsf{K}_2\mathsf{R}_{21}\mathsf{K}_1^{-1} \tag{5}$$

and the two semi-metric cameras are under pure-translating, that means $\mathsf{R}_{21} = \mathtt{I}$. It gives us

$$\omega_{sm2} = \mathsf{T}^{-\top}\omega_{sm1}\mathsf{T}^{-1}$$

where ω_{sm1} and ω_{sm2} are IACs of the two semi-metric cameras.

Applying conic transformation given as (3) makes

$$\boldsymbol{\omega}_2 = \mathsf{H}_{sm2}^\top\mathsf{T}^{-\top}\mathsf{H}_{sm1}^{-\top}\boldsymbol{\omega}_1\mathsf{H}_{sm1}^{-1}\mathsf{T}^{-1}\mathsf{H}_{sm2},$$

and because the infinite homography H_∞^{12} from view 1 and view 2 transforms $\boldsymbol{\omega}_1$ to $\boldsymbol{\omega}_2$, the infinite homography is

$$\mathsf{H}_\infty^{12} = \mathsf{H}_{sm2}^{-1}\mathsf{T}\mathsf{H}_{sm1}. \tag{6}$$

This means that the infinite homography is expressed with semi-metric warping matrices H_{sm1} and H_{sm2} and the infinite homography T between two semi-metric cameras. Note that there are no camera assumptions such as static camera or zero-skew.

3.2 Linear Estimation of Infinite Homography

If a captured scene contains two arbitrary rectangles with an unknown aspect ratio, then the infinite homography is estimated linearly using the parameterization in (6).

Assuming that there are two views that contain two arbitrary rectangles named rectangle i and j, then we can find two infinite homographies with respect to two rectangles as

$$\mathsf{H}^{12}_{\infty,i} = \mathsf{H}^{-1}_{sm2,i}\mathsf{T}_i\mathsf{H}_{sm1,i}$$
$$\mathsf{H}^{12}_{\infty,j} = \mathsf{H}^{-1}_{sm2,j}\mathsf{T}_j\mathsf{H}_{sm1,j}$$

where $\mathsf{H}_{sm1,i}$ means a semi-metric warping matrix of view 1 w.r.t. the rectangle i.

However, the infinite homography is dependent only on the intrinsic parameters of the cameras and the relative rotation between two views. This means that the infinite homography is defined identically regardless of selecting which rectangle is used as a reference. This gives us a constraint equation of:

$$\rho\mathsf{H}^{-1}_{sm2,i}\mathsf{T}_i\mathsf{H}_{sm1,i} = \mathsf{H}^{-1}_{sm2,j}\mathsf{T}_j\mathsf{H}_{sm1,j} \quad (7)$$

where ρ is a proper scale factor.

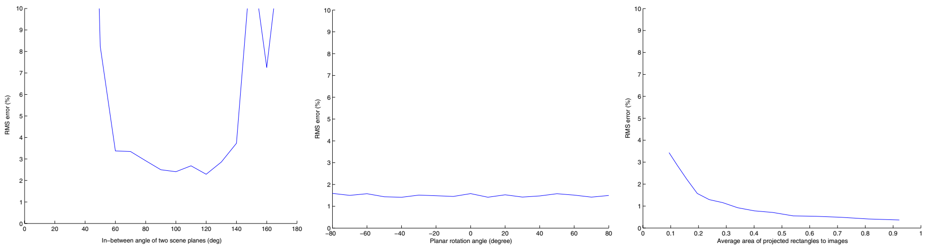
The unknowns are the parameters of T_i and T_j and a scale factor ρ. The number of unknowns is 7 and we have 9 equations, therefore we can easily solve the equation linearly. Note that we do not use any metric measurements, such as lengths or aspect ratios of the scene rectangles.

4 Application to Autocalibration

One of the most important applications of infinite homography is autocalibration of cameras [1]. If the infinite homographies between views captured by a static camera is known, then calibration can be possible linearly without any assumptions on cameras. We applied our proposed algorithm to the autocalibration of a static camera in order to provide validation.

The algorithm to build auto-calibration is as follows.

1. Track two arbitrary rectangles.
2. Find semi-metric warping matrices in all views w.r.t. the two rectangles.
3. Estimate proper transformation T_i and T_j between semi-metric space using (7).
4. Calculate the infinite homography H^{12}_{∞} with semi-metric transformation matrices and obtained proper transformation using (6).
5. Normalize the matrix so that $\det \mathsf{H}^{12}_{\infty} = 1$
6. Find the IAC using $\boldsymbol{\omega} = (\mathsf{H}^{12}_{\infty})^{-\top}\boldsymbol{\omega}(\mathsf{H}^{12}_{\infty})^{-1}$.
7. Determine the camera matrix K from the Cholesky decomposition $\boldsymbol{\omega} = (\mathsf{K}\mathsf{K}^{\top})^{-1}$.



(a) Pose differences **(b)** Planar rotation **(c)** Area

Fig. 3. Simulated performance of the proposed algorithm

This algorithm can be compared with previous works that uses information on scene geometry and proper camera assumptions [9, 8, 10, 11]. The key difference is that ours does not require any metric measurements from the scene, such as line lengths or aspect ratios of the rectangles. Furthermore, our algorithm does not contain camera assumptions, such as zero-skew or known aspect ratio of the pixels. Because it can be much easier to find some rectangles than to find some metrics in images, the proposed method is much more flexible than those given in the previous works.

We first analyzed the performance of the algorithm in various situations. We generated three views with two arbitrary rectangles in general poses and added Gaussian noises with a standard deviation of 0.5 to the corner of the rectangles. Fig. 3 depicts RMS errors of estimated focal length for 500 iterations. Fig. 3a shows the performance to pose differences between two planes in 3D. As expected, the algorithm become singular, when the in-between angle approaches to zero and 180 degrees, since it means the two rectangles are on an identical plane. In 40 degrees, one of the plane is orthogonal to the image plane, and all the features lie on a line. This is a singular case, and in other situation, the calibration is not much degraded for about 90 degrees. Fig. 3b shows the effects of the planar rotation of the world plane. We conclude that the direction of the model axis does not affect the performance of the algorithm. Fig. 3c shows the performance relative to the area of the rectangles used in the images. As expected, the performance of the algorithm increases with the rectangle size.

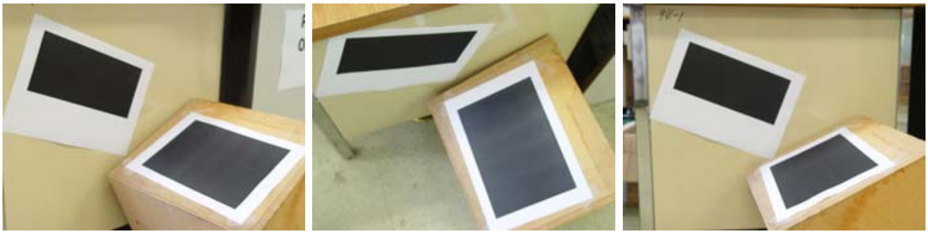


Fig. 4. Input images for auto-calibration using proposed method

The algorithm works well as long as the projected rectangles are larger than 10% of the whole images.

We next applied the algorithm to real images. Fig. 4 shows input images containing two arbitrary rectangles. The images were captured with a SONY DSC-F717 camera in 640 × 480 resolution. The exact values of the aspect ratios of the rectangles are unknown. Since the rectangles are placed arbitrarily, we cannot use the relative pose between two planes. Note that some imaged rectangles are rarely distorted projectively, so we cannot find the vanishing points or lines explicitly.

The estimated infinite homographies are

$$\mathsf{H}_{\infty}^{12} = \begin{bmatrix} 1.0406 & -0.0161 & -208.2218 \\ -0.0167 & 0.2692 & 864.6719 \\ 0.0004 & -0.0009 & 0.6885 \end{bmatrix},$$

$$\mathsf{H}_{\infty}^{12} = \begin{bmatrix} 1.0406 & -0.0161 & -208.2218 \\ -0.0167 & 0.2692 & 864.6719 \\ 0.0004 & -0.0009 & 0.6885 \end{bmatrix}$$

and

$$\mathsf{H}_{\infty}^{13} = \begin{bmatrix} 0.9991 & 0.1037 & -621.4391 \\ 0.0115 & 1.0388 & -127.6288 \\ 0.0006 & 0.0002 & 0.5807 \end{bmatrix}.$$

From the estimated infinite homographies, the intrinsic parameters of the camera is estimated as

$$\mathsf{K}_{estimated} = \begin{bmatrix} 899.4727 & 20.9762 & 322.9044 \\ 0 & 913.2549 & 297.9821 \\ 0 & 0 & 1.0000 \end{bmatrix}.$$

For comparison, we calibrated the camera using the well-known Zhang's plane based calibration method [12] with six metric planes as

$$\mathsf{K}_{Zhang} = \begin{bmatrix} 888.5763 & 14.3200 & 269.8877 \\ 0 & 887.2853 & 243.0086 \\ 0 & 0 & 1.0000 \end{bmatrix}.$$

Note that the proposed algorithm is a kind of autocalibration and does not require any kind of metric measurements, such as metric coordinates or line lengths. Furthermore, although we did not apply any robust method or refinement techniques such as non-linear minimization, the estimated camera parameters are comparable with only three images.

Figure 5 shows another real images captured with the SONY DSC-F717 camera. The yellow lines show the manually selected projected rectangles. The selected rectangles have different aspect ratios and the metric properties of each are unknown. We used only three images captured from different positions. Note

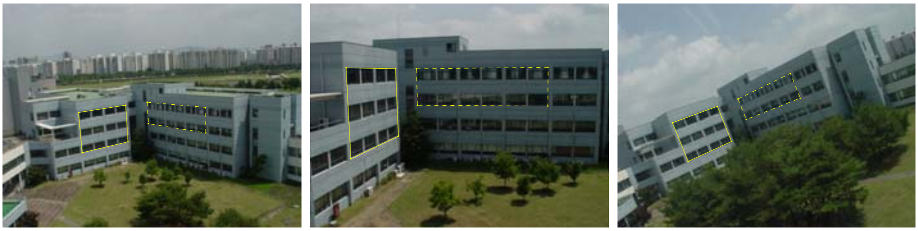


Fig. 5. Another input images for auto-calibration using proposed method

that there are little projective distortions on some projected rectangles. The estimated camera matrix is

$$\mathsf{K}_{estimated} = \begin{bmatrix} 728.5874 & 28.4393 & 352.8952 \\ 0 & 718.3721 & 285.1658 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

and a result from Zhang's calibration method [12] with six metric planes is

$$\mathsf{K}_{Zhang} = \begin{bmatrix} 721.3052 & 2.7013 & 335.3498 \\ 0 & 724.9379 & 247.3248 \\ 0 & 0 & 1.0000 \end{bmatrix} .$$

This shows that autocalibration by our new proposed method can be applicable to real cameras by simply tracking two arbitrary rectangles in general poses.

5 Conclusion

In this paper, we have proposed a new method to estimate the infinite homography from images containing two arbitrary planar rectangles. The proposed method does not require any metric measurements, such as line lengths or aspect ratios of the rectangles.

To deal with rectangles efficiently, we introduce the concept of the semi-metric camera. Semi-metric cameras can be expressed with very simple forms of zero-skew cameras which are related with the aspect ratio of the scene rectangles and camera poses. Also the semi-metric cameras from general views that see an identical 3D rectangles can be regarded as pure-translating cameras. Using this formulation, an infinite homography between two views is expressed simply with semi-metric warping matrices and infinite homography between two semi-metric cameras. Using the fact that the infinite homographies derived from two different rectangles have to be identical, the unknown transformations are estimated linearly.

To validate our method, we used the proposed algorithm for autocalibration of a static camera. The autocalibration results obtained with our novel method were similar to those obtained with well-known plane-based calibration methods, even though our method required only four points on each rectangle and no further refinement.

Acknowledgments. This research has been partially supported by the Korean Ministry of Science and Technology for NRL Program(Grant number M1-0302-00-0064) and by Microsoft Research Asia.

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge (2000)
2. Criminisi, A., Reid, I., Zisserman, A.: Duality, rigidity and planar parallax. In Proceedings of European Conference on Computer Vision. (1998) II: 846
3. Rother, C., Carlsson, S.: Linear multi view reconstruction and camera recovery using a reference plane. International Journal of Computer Vision **49** (2002) 117–141
4. Irani, M., Anandan, P., Weinshall, D.: From reference frames to reference planes: Multi-view parallax geometry and applications. In Proceedings of European Conference on Computer Vision. (1998) II: 829
5. Irani, M., Anandan, P., Cohen, M.: Direct recovery of planar-parallax from multiple frames. IEEE Trans. Pattern Analysis and Machine Intelligence **24** (2002) 1528–1534
6. Pollefeys, M., Van Gool, L.: Stratified self-calibration with the modulus constraint. IEEE Trans. Pattern Analysis and Machine Intelligence **21** (1999) 707–724
7. Kim, J.-S., Kweon, I.S.: Semi-metric space: A new approach to treat orthogonality and parallelism. In Proceedings of Asian Conference on Computer Vision. (2005) To appear
8. Liebowitz, D., Zisserman, A.: Combining scene and auto-calibration constraints. In Proceedings of International Conference on Computer Vision. (1999) 293–300
9. Caprile, B., Torre, V.: Using vanishing points for camera calibration. International Journal of Computer Vision **4** (1990) 127–140
10. Liebowitz, D.: Camera Calibration and Reconstruction of Geometry from Images. PhD thesis, University of Oxford (2001)
11. Wilczkowiak, M., Boyer, E., Sturm, P.: Camera calibration and 3d reconstruction from single images using parallelepipeds. In Proceedings of International Conference on Computer Vision. (2001) I: 142–148
12. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Analysis and Machine Intelligence **22** (2000) 1330–1334

Shape Orientability

Joviša Žunić[1], Paul L. Rosin[2], and Lazar Kopanja[3]

[1] Computer Science Department, Exeter University, Exeter EX4 4QF, UK
J.Zunic@exeter.ac.uk

[2] School of Computer Science, Cardiff University, Cardiff CF24 3AA, UK
Paul.Rosin@cs.cf.ac.uk

[3] Department of Mathematics and Informatics, Novi Sad University,
Trg D. Obradovica 4, 21000 Novi Sad, Serbia and Montenegro
Kopanjal@yahoo.com

Abstract. In this paper we consider some questions related to the orientation of shapes. We introduce as a new shape feature *shape orientability*, i.e. the degree to which a shape has distinct (but not necessarily unique) orientation. A new method is described for measuring shape orientability, and has several desirable properties. In particular, unlike the standard moment based measure of elongation, it is able to differentiate between the varying levels of orientability of n-fold rotationally symmetric shapes.

1 Introduction

This paper deals with some of the problems, and proposes solutions, related to shape *orientability* – i.e. the degree to which shape has distinct (but not necessarily unique) orientation. The computation of a shape's orientation is a common task in the area of computer vision and image processing, being used for example to define a local frame of reference, and helpful for recognition and registration, robot manipulation, etc.

There are situations (see Fig. 1) when the orientation of the shapes seems to be easily and naturally determined. On the other hand, a planar disc could be understood as a shape without orientation.

Most situations are somewhere in between. For very non-regular shapes it could be difficult to say what the orientation should be. Rotationally symmetric polygons could also have poorly defined orientation – see Fig 2 (d). Moreover, even for regular polygons (see Fig. 2 (a) and (b)) is debatable whether they are orientable or not. For instance, is a square an orientable shape? The same question arises for any regular n-gon, but also for shapes having several axes of symmetry, and n-fold ($n > 2$) rotational symmetric shapes. If the answer is "yes, those shapes are somehow orientable", how should the shapes from Fig. 2 be ranked with respect to their orientability? This question is of interest and applicable in the area of shape analysis and shape classification.

The most standard method for computing shape orientability (derived in section 2 and specified in eqn (5)) is based on computing the axis of the least second moment. It is naturally defined and easy to compute. However, it does not specify what the shape orientation should be in those examples (see section 2).

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 11–20, 2006.

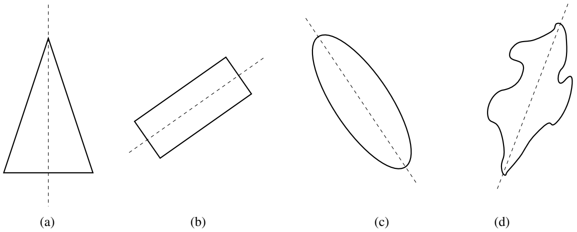


Fig. 1. Reasonable orientations of the shapes coincide with the dashed lines

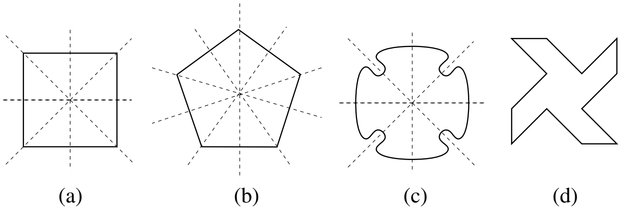


Fig. 2. Reasonable orientations of the shapes coincide with the dashed lines

The problem becomes more complex taking into account that in computer vision and image processing tasks real shapes are replaced with their digitizations. Some specific problems arise when working with digital shapes. Let us mention just two of them:

– Due to the digitization process some "non-orientable" objects may have digitizations whose orientation can be easily computed if (5) is applied.
– On the other hand, it is also possible that some orientable objects have digitalizations which are not orientable.

The impact of digitization effects on changing the computed shape orientation is illustrated by the example of a digitized disc and a digitized square. Even though real discs and squares are not "orientable" shapes (if the standard method is applied – see Lemma 1) it could happen that after digitization, the obtained discrete point sets have an orientation computable in the standard manner. We demonstrate that the computed orientation could depend strongly on:

(a) shape position with respect to the digitization grid;
(b) applied picture resolution.

The effect of item (a) is illustrated by Fig. 3. The same disc is translated into 3 different positions and then digitized. The orientation of the digital disc is not well-defined (in the sense of (5)) for the position displayed in Fig. 3 (a) while the digital discs displayed at Fig. 3 (b) and (c) have the measured orientation $\varphi = \pi/2$ – if (5) is applied. If the applied picture resolution is higher (or equivalently, a bigger disc is digitized) then the impact of the disc position to the computed orientation is higher, as well. As an illustration: we have digitized 16 real discs having the radius equal to 10, whose center positions have been chosen randomly.

For each choice of center position the computed orientations of the obtained digital disc (applying formula (5)) (in the range $[-\pi/2, \pi/2]$) are

0.05	0.03	-0.06	-0.59	0.75	-0.01	-0.23	-0.72
0.13	0.00	0.22	-0.57	-0.06	0.29	-0.61	0.63

and show that the computed orientation strongly depends on the disc position with respect to the digitization grid.

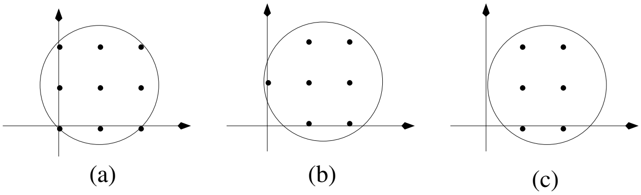


Fig. 3. Three of the 6 non-isometric digitizations of a disc having the radius $\sqrt{2}$ on a binary picture with resolution 1 (i.e., one pixel per measure unit)

Similar problems to the above ones can be caused by noise effects, as well. For instance, consider a square aligned with the coordinate axes. As mentioned, the standard method does not give any answer what the orientation of such a square should be. Adding a single protruding pixel to the boundary can cause the computed orientation to lie anywhere in the range $[-\pi/2, \pi/2]$ depending on its location. As an example, for a 10×10 grid of pixels adding one pixel to the horizontal or vertical edge gives the following computed orientations

0.88	1.00	1.14	1.30	1.48	-1.48	-1.30	-1.14	-1.00	-0.88
-0.69	-0.57	-0.43	-0.27	-0.09	0.09	0.27	0.43	0.57	0.69.

In order to avoid the previously mentioned problems it is not enough to determine if the orientation can be computed or not. It would be useful to see how stable the solution is. For this purpose we will define the *shape orientability* as a shape descriptor. The main purpose of it is to suggest an answer to the question: *Is the computed orientation just a consequence of digitization or noise effects or is it an inherent property of the considered shape?* The orientability can also be used as a shape descriptor in shape classification tasks.

In this paper we will define an orientability measure, which is a number from [0, 1). The defined orientability measure says that a circle has the lowest measured orientability equal to 0. Also, there is no a shape with the measured orientability equal to 1, but shapes having the measured orientability arbitrarily close to 1 can be constructed easily. For example, a rectangle with the edge lengths 1 and a has orientability tending to 1 if $a \to \infty$. This new measure will be described in Section 3. Some experimental results are shown in Section 4, while Section 5 contains concluding remarks.

2 Standard Method

In this section we give a short overview of the method which is mostly used in practice and give a lemma that shows that this method can not be understood as efficient when applied to shapes that have several axes of symmetry.

The standard approach defines the orientation by the so called axis of the least second moment ([3, 4]). That is the line which minimises the integral of the squares of distances of the points (belonging to the shape) to the line. The integral is

$$I(S,\varphi,\rho) = \iint_S r^2(x,y,\varphi,\rho)dxdy \tag{1}$$

where $r(x,y,\varphi,\rho)$ is the perpendicular distance from the point (x,y) to the line given in the form

$$x \cdot \cos\varphi - y \cdot \sin\varphi = \rho.$$

It can be shown that the line that minimizes $I(S,\rho,\varphi)$ passes through the centroid $(x_c(S), y_c(S))$ of the shape S where $(x_c(S), y_c(S)) = \left(\frac{\iint_S xdxdy}{\iint_S dxdy}, \frac{\iint_s ydxdy}{\iint_S dxdy}\right)$. In other words, without loss of generality, we can assume that the origin is placed into the centroid, but also, that the required line minimizing $I(S,\rho,\varphi)$, passes through the origin – i.e., we can set $\rho = 0$. In this way, the shape orientation problem can be reformulated to the problem of determining φ for which the function $F(\varphi, S)$ defined as

$$F(\varphi,S) = I(S,\varphi,\rho=0) = \iint_S (x\cdot\sin\varphi - y\cdot\cos\varphi)^2 dxdy^1$$

reaches the minimum. Once again, we assume that the origin coincides with the center of gravity of S.

Further, if the central geometric moments $\overline{m}_{p,q}(S)$ are defined as usual by:

$$\overline{m}_{p,q}(S) = \iint_S (x - x_c(S))^p \cdot (y - y_c(S))^q \, dx \, dy,$$

and since $(x_c(S), y_c(S)) = (0,0)$ is assumed, we have

$$\begin{aligned} F(\varphi,S) \; &= \; (\sin\varphi)^2 \cdot \overline{m}_{2,0}(S) - \sin(2\cdot\varphi)\cdot\overline{m}_{1,1,}(S) \\ &\quad +(\cos\varphi)^2 \cdot \overline{m}_{0,2}(S). \end{aligned} \tag{2}$$

The minimum of the function $F(\varphi,S)$ can be computed easily. Setting the first derivative $F'(x,S)$ to zero, we have

$$F'(\varphi,S) = \sin(2\varphi)\cdot(\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S)) - 2\cdot\cos(2\varphi)\cdot\overline{m}_{1,1}(S) = 0.$$

[1] The squared distance of a point (x,y) to the line $X\cdot\cos\varphi - Y\cdot\sin\varphi = 0$ is $(x\sin\varphi - y\cos\varphi)^2$.

That easily gives that the required angle φ, but also the angle $\varphi + \pi/2$, satisfies the equation

$$\frac{\sin(2\varphi)}{\cos(2\varphi)} = \frac{2 \cdot \overline{m}_{1,1}(S)}{\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S)}. \tag{3}$$

Consequently, the maximum and minimum of $F(S,\varphi)$ are as follows

$$\max\{F(S,\varphi) \mid \varphi \in [0, 2\cdot\pi]\} = \frac{1}{2} \cdot (\overline{m}_{2,0}(S) + \overline{m}_{0,2}(S))$$

$$+\frac{1}{2} \cdot \sqrt{4 \cdot \overline{m}_{1,1}(S) + (\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S))^2},$$

and

$$\min\{F(S,\varphi) \mid \varphi \in [0, 2\cdot\pi]\} = \frac{1}{2} \cdot (\overline{m}_{2,0}(S) + \overline{m}_{0,2}(S))$$

$$-\frac{1}{2} \cdot \sqrt{4 \cdot \overline{m}_{1,1}(S) + (\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S))^2}.$$

The ratio between $\max\limits_{\varphi\in[0,\pi)} F(S,\varphi)$ and $\min\limits_{\varphi\in[0,\pi)} F(S,\varphi)$

$$\mathcal{E}(S) = \frac{\max\{F(S,\varphi) \mid \varphi \in [0, 2\cdot\pi]\}}{\min\{F(S,\varphi) \mid \varphi \in [0, 2\cdot\pi]\}} \tag{4}$$

is well known as the *elongation* of the shape S.

Let us mention that, when working with digital objects which are actually digitalizations of real shapes, then central geometric moments $\overline{m}_{p,q}(S)$ are replaced with their discrete analogue, i.e., with so called, *central discrete moments.* Since the digitization on the integer grid $\mathbf{Z}^2$ of a real shape S consists of all pixels whose centers are inside S it is natural to approximate $\overline{m}_{p,q}(S)$ by the central discrete moment $\overline{\mu}_{p,q}(S)$ which is defined as

$$\overline{\mu}_{p,q}(S) = \sum_{(i,j)\in S\cap\mathbf{Z}^2} (i - x_{cd}(S))^p \cdot (j - y_{cd}(S))^q\,,$$

where $(x_{cd}(S), y_{cd}(S)) = \left(\frac{\sum\limits_{(x,y)\in S\cap\mathbf{Z}^2} x}{\sum\limits_{(x,y)\in S\cap\mathbf{Z}^2} 1}, \frac{\sum\limits_{(x,y)\in S\cap\mathbf{Z}^2} y}{\sum\limits_{(x,y)\in S\cap\mathbf{Z}^2} 1} \right)$ is the centroid of discrete shape $S \cap \mathbf{Z}^2$.

Some answers about the efficiency of the approximation $\overline{m}_{p,q}(S) \approx \overline{\mu}_{p,q}(S)$ can be found in [5].

If the geometric moments in (3) are replaced with the corresponding discrete moments we have the equation

$$\frac{\sin(2\varphi)}{\cos(2\varphi)} = \frac{2 \cdot \overline{\mu}_{1,1}(S)}{\overline{\mu}_{2,0}(S) - \overline{\mu}_{0,2}(S)} \tag{5}$$

which describes the angle φ which is used as an approximate orientation of the shape S, i.e., the angle which is used to describe the orientation of discrete shape

$S\cap\mathbf{Z}^2$. It is worth noting that equation (5) can be derived easily if the orientation of the discrete set (a finite number point set) $S\cap\mathbf{Z}^2$ is defined by the line (passing the origin) which minimizes the total sum $\sum_{(i,j)\in S\cap\mathbf{Z}^2}(i\cdot\sin\varphi - j\cdot\cos\varphi)^2)$ of squares of distances of points from $S\cap\mathbf{Z}^2$ to this line.

In other words, the equality (5) can be derived as a consequence when trying to solve the following optimization problem

$$\min\left\{\sum_{(i,j)\in S\cap\mathbf{Z}^2}(i\cdot\sin\varphi - j\cdot\cos\varphi)^2 \mid \varphi\in[0,\pi]\right\} \tag{6}$$

assuming that the centroid $\left(x_{cd}(S\cap\mathbf{Z}^2),\ y_{cd}(S\cap\mathbf{Z}^2)\right)$ coincides with the origin.

So, the standard method is very simple (in both "real" and "discrete" versions) and it comes from a natural definition of the shape orientation. However, it is not always effective. The next lemma shows that the method does not always give a clear answer what the shape orientation should be – for more details see [9].

Lemma 1. *If a given shape S has more than two axes of symmetry then $F(\varphi,S)$ is a constant function.*

Proof. From (3) it is obvious that the function $F(\varphi,S)$ could have exactly one maximum and one minimum on the interval $[0,\pi)$, or it must be a constant function. Trivially $F(0,S)=F(\pi,S)$. So, if S has more than two axes of symmetry it must be constant since $F'(\varphi,S)$ does not have more than two zeros on the interval $[0,\pi)$. ▯

Remark. A direct consequence of Lemma 1 is that

- $F(S,\varphi)=\frac{1}{2}\cdot(\overline{m}_{2,0}(S)+\overline{m}_{0,2}(S))$ for all $\varphi\in[0,\pi)$;
- $\mathcal{E}(S)=1$

holds for all shapes that have more than two axes of symmetry. In other words, the standard method does not specify the orientation of shapes from Fig. 2, or more generally, what the orientation is for shapes having more than two axes of symmetry. Also, for all such shapes the measured elongation is 1 – i.e., the same as the measured elongation for a circle, what is not a desirable property.

3 Measuring Shape Orientability

In this section we consider what quantity can be used to describe shape orientability – to be used as an inherent shape property.

Intuitively, it can be assumed that shapes with high measured elongation are more orientable than shapes with lower measured elongation. Thus, the elongation $\mathcal{E}(S)$ (see (4)) can be used to estimate shape orientability. Since $\mathcal{E}(S)\in[1,\infty)$, in order to have the measured orientability between 0 and 1, we can measure the orientability as:

$$1-\frac{1}{\mathcal{E}(S)}. \tag{7}$$

Several other measures can be derived from the function $F(S, \varphi)$, as well. For example, a larger ratio between the areas of the regions bounded by:

- the coordinate axes, line $y = \min\limits_{\varphi \in [0,\pi)} F(S, \varphi)$, and line $x = \pi$, and
- the coordinate axes, line $y = F(S, \varphi)$, and line $x = \pi$,

should indicate a lower shape orientability. This leads to the following:

Definition 1. *For a given shape S its orientability $\mathcal{D}_F(S)$ can be measured as*

$$\mathcal{D}_F(S) = 1 - \frac{\pi \cdot \min\{F(S,\varphi) \mid \varphi \in [0,\pi)\}}{\int_0^\pi F(S,\varphi) \cdot d\varphi}$$

$$= \frac{\sqrt{4 \cdot (\overline{m}_{1,1}(S))^2 \ + \ (\overline{m}_{2,0}(S) - \overline{m}_{0,2}(S))^2}}{\overline{m}_{2,0}(S) + \overline{m}_{0,2}(S)}.$$

Obviously, $\mathcal{D}_F(S)$ is easily computable and well-motivated. However, it is clear that all shape orientability measures based on $F(S, \varphi)$ are limited by the result of Lemma 1, i.e., $\mathcal{D}_F(S) = 1 - 1/\mathcal{E}(S) = 0$ for all shapes S having more than two axes of symmetry. In some situations (applications) a new measure for shape orientability is required that does not have that disadvantage.

Now, we define such a measure. When dealing with shapes that have several axes of symmetry, such shapes do not necessarily have identical measured orientability, as would result when using $1 - 1/\mathcal{E}(S)$ and $\mathcal{D}_F(S)$, for example.

Definition 2. *For a given shape S let $R(\alpha)$ be the minimal rectangle whose edges make an angle α with the coordinate axes and which includes S (see Fig. 4). Let the following hold:*

$$\mathcal{A}_{min}(S) \ = \ \min_{\alpha \in [0,\pi)} \{ \ Area\_of\_R(\alpha) \ \},$$

$$\mathcal{A}_{max}(S) \ = \ \max_{\alpha \in [0,\pi)} \{ \ Area\_of\_R(\alpha) \ \}.$$

Then, we define the orientability measure $\mathcal{D}(S)$ of the shape S as:

$$\mathcal{D}(S) \ = \ 1 - \frac{\mathcal{A}_{min}(S)}{\mathcal{A}_{max}(S)}.$$

The next theorem describes some desirable properties of $\mathcal{D}(S)$. Because of simplicity, the proof is omitted.

Theorem 1. *The new defined measure for the shape orientability has the following properties:*

- $\mathcal{D}(S) \in [0, 1)$ *for any shape S;*
- *A circle has the measured orientability equal to* 0*;*
- *The measured orientability is invariant w.r.t. similarity transformations.*

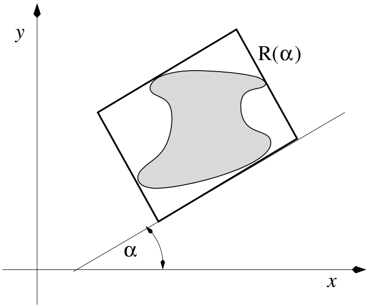


Fig. 4. The rectangle $R(\alpha)$ is the minimum area rectangle which includes the given shape (dashed area) and whose edges make an angle α with the coordinate axes

The new orientability measure introduced by Definition 2 is very convenient for numerical computation with arbitrary precision. The exact computation of $\mathcal{D}(S)$ when the measured shape S is a polygon will be described in detail in a forthcoming publication by the authors. Note that the problem of computation of $\mathcal{A}_{min}(S)$ is well studied in literature. It has been shown [2] that for a given polygon S a rectangle which has the minimal possible area and which includes the polygon S must have an edge parallel to an edge of the convex hull of S. An efficient, linear time, algorithm for such a computation (if S is a simple polygon) has been described in [8], using the technique of orthogonal calipers.

The main objection to $\mathcal{D}(S)$ is that shapes having the same convex hull have the same measured orientability. The following slight modification of Definition 2 ensures that a given non-convex shape does not have the measured orientability equal to the measured orientability of its convex hull.

Definition 3. *For a given shape S let $R(\alpha)$, $\mathcal{A}_{min}(S)$, and $\mathcal{A}_{max}(S)$ be defined as in Definition 2. Then, for any real number $\alpha \in [0,1)$ we define the orientability measure $\mathcal{D}_\alpha(S)$ of the shape S as:*

$$\mathcal{D}_\alpha(S) \;=\; 1 - \frac{\mathcal{A}_{min}(S) - \alpha \cdot Area\_of\_S}{\mathcal{A}_{max}(S) - \alpha \cdot Area\_of\_S}.$$

Note that the orientability measure $\mathcal{D}_\alpha$ also has the desirable properties listed in Theorem 1.

4 Some Examples

We now give some examples of orientability calculated using the new measure. The first example (see Fig. 5) shows synthetic data, mostly exhibiting both rotational and reflectional symmetries. Theory tells us that $\mathcal{D}_F(S)$ should produce values of zero; in practice quantization errors have caused non-symmetries, but the values remain close to zero. The fourth shape in Fig. 5 (a) has only one axis of symmetry; nevertheless, since the indentation in the square has a relatively

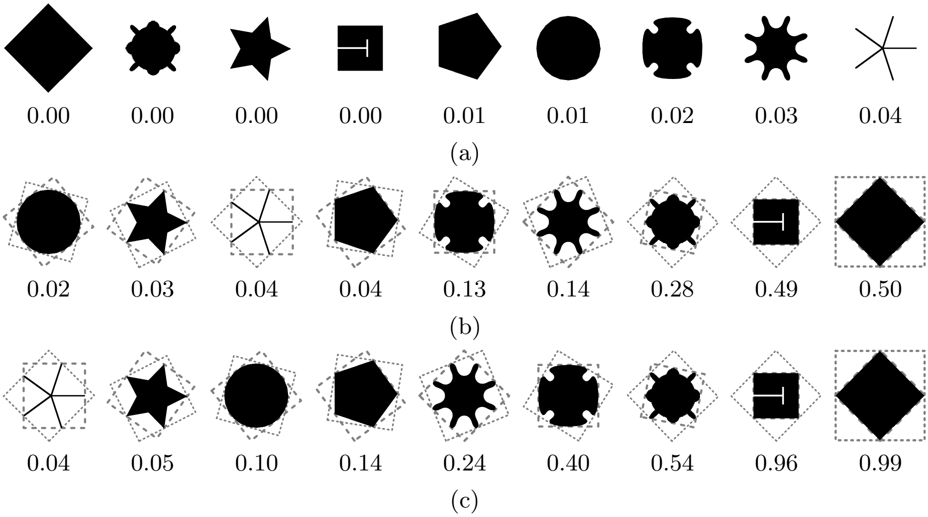


Fig. 5. Synthetic data ordered by orientability using a) $\mathcal{D}_F(S)$, b) $\mathcal{D}(S)$, c) $\mathcal{D}_{\alpha=1}(S)$. The rectangles corresponding to $\mathcal{A}_{min}$ (dashed) and $\mathcal{A}_{max}$ (dotted) are overlaid.

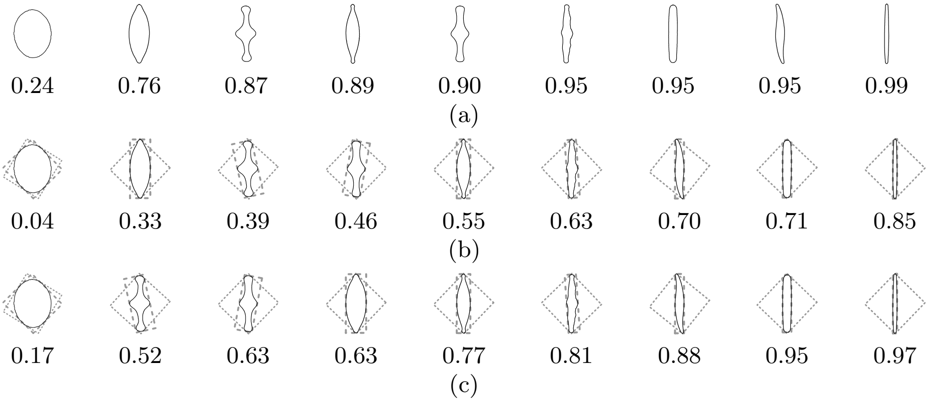


Fig. 6. Diatom data ordered by orientability using a) $\mathcal{D}_F(S)$, b) $\mathcal{D}(S)$, c) $\mathcal{D}_1(S)$

small area it does not substantially affect the values of the moments, and therefore $\mathcal{D}_F(S)$ is approximately zero. In contrast to $\mathcal{D}_F(S)$, $\mathcal{D}(S)$ does differentiate between the shapes. Again, according to theory, the first shape in Fig. 5b that looks like a circle, but is actually a 24-gon, is assigned a value close to zero.

The second set of examples (see Fig. 6) consists of the outlines of *diatoms* – unicellular water borne algae used previously by Žunić and Rosin [10] in the development of convexity measures. Future work will look at applying the orientability measure to classifying the diatoms, as in [10].

5 Concluding Remarks

We have defined shape orientability as a new shape descriptor. We also discuss some approaches for measuring shape orientability and define a new measure. The purpose of such a measure is to give an answer as to whether the computed orientation of a shape is an inherent property of the considered shape, or whether it comes from artifacts caused by the digitization process or by noise, for example. The measure can be useful if applied to shapes whose measured orientation changes even under slight deformations [1].

The shape orientability measured by the method presented here is a number in the form $[0, 1)$. The minimal possible measured orientability (equal to zero) is for a disc. There is no shape with a measured orientability equal to 1. Even in cases where there is no doubt what the orientation should be, e.g. an elongated rectangle, the measured orientability is not 1. That could be desirable property because the measured orientation for rectangles increases if the ratio between length a of the longer edge and the length b of the shorter edge increases as well. In the limit case when a is a positive constant while $b \to 0$, the measured orientability tends to 1 and we could say that a straight line segment is a perfectly oriented shape. Another desirable property is that the shapes with several axes of symmetry could have non-zero measured orientability. As an illustration, a regular $4n$-gon P_{4n} has the measured orientability $\mathcal{D}(P_{4n})$ equal to $\mathcal{D}(P_{4n}) = 1 - \frac{4\cos\frac{\pi}{4n}}{4} = 1 - \cos\frac{\pi}{4n}$. Obviously, $\mathcal{D}(P_{4n})$ tends to 0 as $n \to \infty$.

References

1. J. Cortadellas, J. Amat, F. de la Torre, "Robust Normalization of Silhouettes for Recognition Application," *Patt. Rec. Lett.,* Vol. 25, pp. 591-601, 2004.
2. H. Freeman, R. Shapira, "Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve," *Comm. of the ACM,* Vol. 18, pp. 409-413, 1975.
3. B. K. P. Horn, *Robot Vision,* MIT Press, 1986.
4. R. Jain, R. Kasturi, B. G. Schunck, *Machine Vision,* McGraw-Hill, 1995.
5. R. Klette, J. Žunić, "Digital approximation of moments of convex regions," *Graphical Models and Image Processing,* Vol. 61, pp. 274-298, 1999.
6. S.E. Palmer, *Vision Science: Photons to Phenomenology,* MIT Press, 1999.
7. F.P. Preparata and M.I. Shamos, *Computational Geometry,* Springer-Verlag, 1985.
8. G.T. Toussaint, "Solving geometric problems with the rotating calipers," *Proc. IEEE MELECON '83*, pages A10.02/1–4, 1983.
9. W.H. Tsai, S.L. Chou, "Detection of Generalized Principal Axes in Rotationally Symmetric Shapes," *Patt. Rec.,* Vol. 24, pp. 95-104, 1991.
10. J. Žunić and Rosin, P.L., "A New Convexity Measurement for Polygons," *IEEE Trans. PAMI,* Vol. 26, No. 7, pp. 923–934, 2004.

How to Compute the Pose of an Object Without a Direct View?

Peter Sturm and Thomas Bonfort

INRIA Rhône-Alpes, 38330 Montbonnot St Martin, France
{Peter.Sturm, Thomas.Bonfort}@inrialpes.fr

Abstract. We consider the task of computing the pose of an object relative to a camera, for the case where the camera has no direct view of the object. This problem was encountered in work on vision-based inspection of specular or shiny surfaces, that is often based on analyzing images of calibration grids or other objects, reflected in such a surface. A natural setup consists thus of a camera and a calibration grid, put side-by-side, i.e. without the camera having a direct view of the grid. A straightforward idea for computing the pose is to place planar mirrors such that the camera sees the calibration grid's reflection. In this paper, we consider this idea, describe geometrical properties of the setup and propose a practical algorithm for the pose computation.

1 Introduction

Consider a calibration grid or any other known object, planar or not, and a camera. We would like to determine their relative pose, but for the case where **the camera does not see the object directly**. This is an unusual setting, but it is quite natural for the task of reconstructing specular or shiny surfaces, as explained in the following. Modeling of specular or shiny surfaces is an important application in inspection of industrial parts, especially in the car manufacturing industry (control of wind shields and bodywork) but also in the control of optical lenses or mirrors, glasses of watches etc. Vision-based reconstruction of specular surfaces is usually based on acquiring images of known patterns or light sources, reflected in the surface to be reconstructed [3, 4, 6, 7, 11, 14].

It is thus rather natural to place the camera and pattern such that the camera does not have a direct view of the latter, or at most sees a small part of it. We have proposed practical approaches for the reconstruction of specular surfaces where such an arrangement is indeed used. The question of how to compute the pose of an object without a direct view is thus important for us and in addition scientifically appealing.

Our initial solution consisted in attaching the pattern rigidly to the camera and to move the two to a few locations. During this, the camera acquired images of a calibration grid, and a secondary camera (static) acquired images of our pattern. With this input, the pattern's 3D trajectory was computed as well as the main camera's one. By registering the two trajectories into a common coordinate frame, along the lines of [2] and of the classical hand-eye calibration

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 21–31, 2006.

problem, we finally computed the pose of the pattern relative to the main camera. This approach was found to be too cumbersome in practice. A second camera is required and especially, having to move the camera–pattern pair is not desirable, as we currently use an LCD monitor to produce the pattern(s).

We are thus aiming at a lighter procedure. A natural idea is to proceed as follows: place a planar mirror in different positions in front of the camera such that the pattern's reflection is seen, and acquire images. The question arises if this input is sufficient to solve our pose problem, and if yes, how many positions of the planar mirror are required? We show in this paper that our pose problem can be solved up to 1 degree of freedom from two positions, and can be fully solved from three or more positions.

2 Background

2.1 Camera Model

We consider perspective projection as camera model. The projection of 3D points is modeled by a 3×4 projection matrix $\mathsf{P} = \mathsf{KR}\left(\mathtt{I} | -\mathbf{t}\right)$, where K is the usual 3×3 calibration matrix with the camera's intrinsic parameters, and the orthogonal matrix R and the vector $\mathbf{t}$ represent camera orientation and position. For simplicity, we assume that the camera is calibrated, i.e. that K is known (this will be relaxed later). We thus directly work with geometric image coordinates, i.e. consider that 3D points $\mathbf{Q}$ are projected to image points $\mathbf{q}$ via the canonical projection matrix $\mathbf{q} \sim \mathsf{R}\left(\mathtt{I} | -\mathbf{t}\right)\mathbf{Q}$. 2D and 3D points are expressed in homogeneous coordinates and $\sim$ means equality of vectors or matrices, up to scale.

2.2 Pose Computation

A classical task of photogrammetry and computer vision is to compute the pose of a calibrated camera, relative to an object of known structure. In this work, we use planar reference objects. There exist many algorithms for the planar pose problem; we use [10].

2.3 Reflections in Planes

Consider a plane $\Pi = \left(\mathbf{n}^{\mathsf{T}}, d\right)^{\mathsf{T}}$ in 3-space, i.e. consisting of points satisfying the equation $n_1 X + n_2 Y + n_3 Z + d = 0$. In the following, we will always suppose that the plane's normal vector is of unit norm. The reflection in Π can be represented by the following transformation matrix:

$$\mathsf{S} = \begin{pmatrix} \mathtt{I} - 2\mathbf{n}\mathbf{n}^{\mathsf{T}} & -2d\mathbf{n} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix}$$

Let us denote the upper left 3×3 matrix of S by $\bar{\mathsf{S}}$. It is an orthogonal matrix, with determinant -1 (whereas a rotation matrix has determinant $+1$). Further, it has $+1$ as double eigenvalue and -1 as single eigenvalue. The plane normal $\mathbf{n}$ is an eigenvector of $\bar{\mathsf{S}}$ to the eigenvalue -1. Note also that $\mathsf{S}^{-1} = \mathsf{S}$.

2.4 Planar Motion and Fixed-Axis Rotation

Planar motion usually means a translation in some direction, followed by a rotation about an axis that is orthogonal to the translation direction. Such a motion can always be expressed as just a rotation about an axis that is parallel to the above rotation axis; we thus prefer to call such motions *fixed-axis rotations.* It is easy to show that any euclidian transformation that preserves some line point-by-point, is a fixed-axis rotation, whose axis is that line.

Let the axis be represented by its direction vector $\mathbf{D}$ and a footpoint $\mathbf{A}$ such that $\mathbf{A} + \lambda\mathbf{D}$ represents the points (in non-homogeneous coordinates) on the axis. Any finite point on the axis can serve as footpoint; we always choose the one that is "orthogonal" to $\mathbf{D}$: $\mathbf{A}^{\mathsf{T}}\mathbf{D} = 0$. This is the point on the axis that is closest to the origin.

Let α be the angle of rotation and R be the rotation matrix representing rotation by α about $\mathbf{D}$. Then, the 4×4 matrix representing the complete fixed-axis rotation, is:

$$\mathsf{T} = \begin{pmatrix} \mathtt{I} & \mathbf{A} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix} \begin{pmatrix} \mathsf{R} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix} \begin{pmatrix} \mathtt{I} & -\mathbf{A} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix} = \begin{pmatrix} \mathsf{R} & \mathbf{A} - \mathsf{R}\mathbf{A} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix} .$$

2.5 Reflection in Two Planes

Consider successive reflections in two planes. It can be shown that this is a fixed-axis rotation, with the intersection line of the two planes as rotation axis: the transformation preserves the intersection line of the two planes point-by-point, and thus is a fixed-axis rotation.

Further, the rotation angle is twice the angle between the two planes. This is also easy to see: let the transformation be the sequence $\mathsf{S}_2\mathsf{S}_1$ of reflections in two planes. Let us apply this transformation to the point at infinity $(\mathbf{n}_1^{\mathsf{T}}, 0)^{\mathsf{T}}$, i.e. the normal direction of the first plane. This is a fixed point of S_1, hence the transformation gives the point's reflection in S_2. The angle between the original point at infinity, and the transformed one, i.e. the fixed-axis rotation angle, is thus twice the angle between the original point at infinity and the second reflection plane. Hence, as said above, the sequence of reflections in two planes is a fixed-axis rotation, whose angle is twice the angle between the planes.

2.6 Horopter

The horopter of a stereo system is the set of 3D points that project to points with identical coordinates in the two cameras. Let P_1 and P_2 be the two cameras' projection matrices. The horopter thus consists of all 3D points $\mathbf{Q}$ with $\mathsf{P}_1\mathbf{Q} \sim \mathsf{P}_2\mathbf{Q}$. This is in general a quartic curve. If the two cameras have identical calibration and are separated by a fixed-axis rotation, then the horopter "degenerates" into the union of a straight line and a circle: the motion's rotation axis and the circle in the motion plane that contains the two optical centers and that cuts the rotation axis [5].

3 Outline of the Proposed Approach

We consider a camera and an object in fixed position, put a planar mirror in the scene in n different positions, and take an image for each of those. We suppose that the camera sees the object's reflection in each image. We further suppose that the object's structure is known and that correspondences between object and image points can be obtained.

In the first step of our approach, the views of the reflected object are treated as if they were direct views. We may thus compute a camera pose, from the camera's calibration and the given point correspondences. This will actually give the pose of a "virtual" camera that is the reflection of the true camera, in the planar mirror (cf. figure 1). Overall, we thus get the pose of n virtual cameras, relative to the object.

In the second step, we try to infer the positions of the planar mirrors. The underlying constraint is that reflecting the virtual cameras in mirrors with the correct positions, will lead to n identical cameras – the true one. We show how the mirror positions can be computed using the above notions of horopter and fixed-axis rotation. We further show that for $n = 2$, the problem can be solved up to 1 degree of freedom, and that with $n > 2$ a unique solution can be found. These steps are described in the following sections.

4 Computing Pose of Virtual Cameras

In the following, we adopt the object's coordinate system as our reference system, in which the pose of true and virtual cameras will be expressed. Let the pose of the true camera be represented by the projection matrix

$$\mathsf{R}\left(\mathtt{I}|-\mathbf{t}\right)$$

Consider now a planar mirror, defined by the plane

$$\Pi = \begin{pmatrix}\mathbf{n}\\ d\end{pmatrix}$$

Object points $\mathbf{Q}$ are projected into the true camera as follows:

$$\mathbf{q} \sim \mathsf{R}\left(\mathtt{I}|-\mathbf{t}\right)\begin{pmatrix}\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T} & -2d\mathbf{n}\\ \mathbf{0}^\mathsf{T} & 1\end{pmatrix}\mathbf{Q}$$

From point correspondences $(\mathbf{Q}, \mathbf{q})$, we can run any pose computation algorithm and compute the projection matrix of the virtual camera:

$$\mathsf{P} = \mathsf{R}\left(\mathtt{I}|-\mathbf{t}\right)\begin{pmatrix}\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T} & -2d\mathbf{n}\\ \mathbf{0}^\mathsf{T} & 1\end{pmatrix} = \mathsf{R}\left(\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T}\right)\left(\mathtt{I}\ 2d\mathbf{n}-\left(\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T}\right)\mathbf{t}\right) \quad (1)$$

One issue needs to be considered: pose algorithms for perspective cameras compute a pose consisting of a rotation and a translation component, whereas the above projection matrix contains a reflection part. What the pose computation will compute is thus a rotation matrix R' and a camera position $\mathbf{t}'$, with:

$$\mathsf{P} \sim \underbrace{\mathsf{R}\left(2\mathbf{n}\mathbf{n}^\mathsf{T}-\mathtt{I}\right)}_{\mathsf{R}'}\left(\mathtt{I}\ -\mathbf{t}'\right) \quad \text{with} \quad -\mathbf{t}' = 2d\mathbf{n}-\left(\mathtt{I}-2\mathbf{n}\mathbf{n}^\mathsf{T}\right)\mathbf{t}$$

Our input for the following steps is thus a set of n projection matrices P_i (we drop the ' above the R_i and $\mathbf{t}_i$):

$$\mathsf{P}_i = \mathsf{R}_i \begin{pmatrix} \mathtt{I} & -\mathbf{t}_i \end{pmatrix}$$

The basic constraint for solving our pose problem is the following (cf. §3): we try to compute n planes Π_i and associated reflection matrices S_i such that

$$\forall i, j : \mathsf{P}_i\mathsf{S}_i \sim \mathsf{P}_j\mathsf{S}_j$$

If there is a unique solution for the set of planes, then any $\mathsf{P}_i\mathsf{S}_i$ gives the pose of the true camera. In the above equation, we may actually replace the equality up to scale by a component-wise equality, since the determinants of the left 3×3 submatrices of the $\mathsf{P}_i\mathsf{S}_i$ are all equal to -1. Hence, our constraint becomes:

$$\forall i, j : \mathsf{P}_i\mathsf{S}_i = \mathsf{P}_j\mathsf{S}_j .$$

5 Two Mirror Positions

In this section, we investigate what can be done from just two mirror positions. Our basic constraint is:

$$\mathsf{P}_1\mathsf{S}_1 = \mathsf{P}_2\mathsf{S}_2$$

Instead of directly trying to compute the reflections S_1 and S_2, we first concentrate on:

$$\mathsf{P}_1 = \mathsf{P}_2\mathsf{S}_2\mathsf{S}_1^{-1} = \mathsf{P}_2\mathsf{S}_2\mathsf{S}_1$$

We have seen above that the sequence of two reflections gives a fixed-axis rotation. Let us thus compute R and $\mathbf{t}$ in the following euclidian transformation between the two virtual cameras:

$$\mathsf{P}_1 = \mathsf{P}_2 \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}$$

We get $\mathsf{R} = \mathsf{R}_2^\mathsf{T}\mathsf{R}_1$ and $\mathbf{t} = \mathbf{t}_2 - \mathsf{R}_2^\mathsf{T}\mathsf{R}_1\mathbf{t}_1$. In the following, we analyze what R and $\mathbf{t}$ reveal about the individual reflections S_1 and S_2.

Let α be the rotation angle of R. We already know that it equals twice the angle between the two mirror planes. Further, we want to compute the fixed axis (the intersection of the two mirror planes). Let us represent it by its direction $\mathbf{D}$ and a footpoint $\mathbf{A}$, cf. §2.4. The direction $\mathbf{D}$ is identical with the rotation axis of R and can for example be computed as its eigenvector to the eigenvalue $+1$. Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be an orthonormal basis of the complement of $\mathbf{D}$, such that:

$$\mathsf{R}\begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 \end{pmatrix} \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

As for the footpoint $\mathbf{A}$, we compute it as follows. Since we want $\mathbf{A}$ to be "orthogonal" to $\mathbf{D}$, we can parameterize it by two scalars a_1 and a_2:

$$\mathbf{A} = \begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

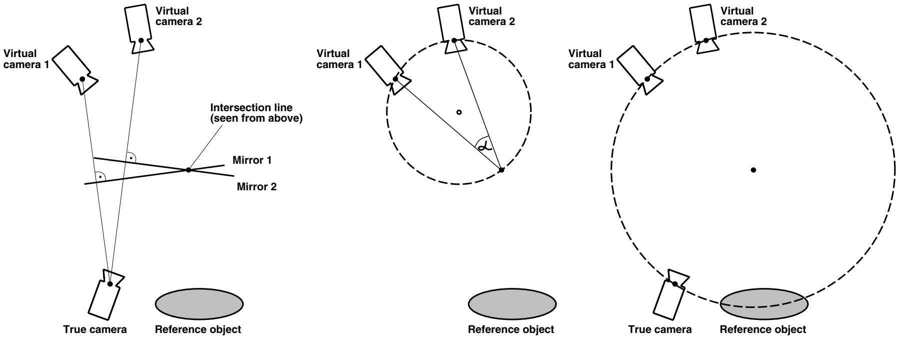


Fig. 1. Illustration of the case of two planar mirrors. **Left:** the virtual cameras are the reflections of the true one in the planar mirrors. **Middle:** the horopter curve of the two virtual cameras is the union of the shown circle and the axis of the fixed-axis rotation, i.e. the mirror planes' intersection axis. Further shown is the angle α of the fixed-axis rotation. **Right:** the true camera pose can be recovered up to one degree of freedom. The reconstructed camera position is only constrained to lie on the shown circle.

The translation part of the fixed-axis rotation would thus be (cf. §2.4):

$$\mathbf{A} - \mathsf{R}\mathbf{A} = \begin{pmatrix}\mathbf{D}_1 & \mathbf{D}_2\end{pmatrix} \begin{pmatrix} 1-\cos\alpha & \sin\alpha \\ -\sin\alpha & 1-\cos\alpha \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

In the absence of noise, this would be equal to $\mathbf{t}$. However, with noise, the computed R and $\mathbf{t}$ will in general not exactly represent a fixed-axis rotation. We thus determine a_1 and a_2 that minimize the L_2 norm of:

$$\mathbf{t} - \begin{pmatrix}\mathbf{D}_1 & \mathbf{D}_2\end{pmatrix} \begin{pmatrix} 1-\cos\alpha & \sin\alpha \\ -\sin\alpha & 1-\cos\alpha \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

This is a linear least squares problem, with the following closed-form solution:

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & -\cot\frac{\alpha}{2} \\ \cot\frac{\alpha}{2} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{D}_1^\mathsf{T} \\ \mathbf{D}_2^\mathsf{T} \end{pmatrix} \mathbf{t}$$

So far, we have computed the axis and angle of the fixed-axis rotation being the sequence of S_2 and S_1. What does this tell us about the mirror planes Π_1 and Π_2? The axis being the planes' intersection line, we know that both planes must contain it; this determines each plane up to a rotation about the axis. Further, we know the angle between the planes ($\alpha/2$). In addition, not explained here in more detail, we know the "ordering" of the two planes, i.e. the second plane has always to be on the same side of the first (in terms of rotation about their intersection line). Overall, we thus have computed the two mirror planes up to a single unknown. It can be shown (not done due to lack of space) that this can not be reduced further with only two planes.

A geometric illustration of the situation is given in figure 1. For simplicity, we show the scene as seen from the direction of the mirror planes' intersection line. On the right, the ambiguity in the inferred pose of the true camera is shown: its position can lie anywhere on the circle that is centered in the fixed-axis rotation axis, is “orthogonal” to the latter and passes through the two virtual camera positions. Let us call this circle the *pose circle*. For every possible camera position on the pose circle though, the camera’s orientation is uniquely defined.

All possible poses for the true camera can be parameterized by an angle β as follows. Any plane containing the axis of the fixed-axis rotation, can be written

$$\Pi \sim \begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 \\ -a_1 & -a_2 \end{pmatrix} \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix}$$

for some β. We can thus parameterize the possible poses of the true camera by β, by reflecting any of the virtual ones, say the first, in the family of planes Π.

6 Three or More Mirror Positions

With three or more mirror positions, our pose problem will in general be solvable. Different approaches are possible. One could for example use the solution of the previous section for all available pairs of mirror positions. The problem could be geometrically expressed as one of finding the common point of a set of circles in 3-space (the circles as sketched in the right part of figure 1). A few special cases need to be discussed:

- Consider the case where the planar mirror is rotated about some axis contained in the mirror plane. In that case, the fixed-axis rotations for pairs of mirror positions will all have the same axis, and the resulting pose circles will all be identical. The pose of the true camera will remain ambiguous. Note that this case also refers to mirror positions that are parallel to each other (this can be seen as rotating the mirror about a line at infinity).
- The case where the mirror moves in such a way that it remains tangent to some cylinder. This implies that the lines of intersection for pairs of positions, will be parallel to one another. Hence, all pose circles lie on the same plane but since we have three or more of them, there will be a single common point: the position of the true camera.
- If there are intersection lines for pairs of mirror positions that are not parallel, then there are pose circles with different supporting planes. It can be shown that there will be pose circles in at least three supporting planes with different normals. Consequently, the set of pose circles can only have a single common point (the intersection point of all supporting planes), meaning that again, the pose problem can be solved.

In the following, we present a less geometrical method for combining results from pairs of mirror positions. Consider mirror position i. Using §5, we can compute the intersection lines of the mirror in that position, with any other position. The plane at position i has to contain all these lines, and can thus be uniquely computed from two or more lines, unless all of them are identical (cf. the above discussion). In the presence of noise, there will not be a plane that exactly contains all these lines, and we perform a fitting procedure, as follows. First, perform a least-squares fit to the direction vectors of all available lines. This will be used as the plane's normal vector. Then, compute the plane position (the scalar d appearing elsewhere in this paper) that minimizes the sum of squared distances to the available footpoints. This method fails if all lines are parallel. An alternative procedure for that case is simple to devise though.

Once mirror plane positions are computed, we reflect the virtual cameras in the corresponding planes, to obtain the true camera's projection matrix. The complete method is summarized in the next section.

7 Complete Approach

1. For each mirror position, compute the pose of the virtual camera, relative to the object.
2. For each pair of mirror positions, compute the fixed-axis rotation between the virtual cameras.
3. For each mirror position, compute the mirror plane by fitting it to the associated axes of fixed-axis rotations.
4. Compute the true camera's projection matrix by reflecting any virtual camera in the associated plane.
5. Do a non-linear bundle adjustment: minimize the sum of squared reprojection errors, over the pose of the true camera and the positions of the mirror plane. This is implemented in the usual sparse manner [12].

For conciseness, we did not mention that in practice we also compute intrinsic camera parameters during this procedure: in the first step, we also calibrate the camera. In practice, we use a planar reference object; we thus use the method of [13, 9] to calibrate the camera from the reflected views (the reflection does not alter the intrinsic parameters), prior to computing pose in step 1. Further, the bundle adjustment in the last step also optimizes intrinsic camera parameters.

8 Experiments

8.1 Setup

We use an LCD monitor as reference object, considering it effectively as a planar surface. A structured light type approach [8] is used to get correspondences between the screen and the image plane: for each position of the planar mirror, we actually take a set of images, with the screen displaying a sequence of different patterns (cf. figure 2). Patterns are designed such that for each pixel in the image plane, we can directly compute the matching "point" on the screen, from the sequence of black-and-white (dark-and-light) greylevels received at the pixel.

8.2 Surface Reconstruction

We tested our method on real images, cf. figure 2. It was difficult to evaluate the estimated pose, so we evaluate it indirectly as follows. In [1], we describe an approach for the reconstruction of general specular surfaces from two images of the reference object's reflections. Here, to perform a quantitative evaluation, we reconstruct a planar specular surface (a hard drive platter), without making use of the planarity information for the reconstruction. Images are taken with a fixed pose of the camera and the specular surface (cf. figure 3), but with two different positions of the LCD monitor. Each of the two positions is estimated using the approach presented in this paper, by placing planar mirrors in the scene and making use of the knowledge of planarity.

The specular surface is reconstructed as a dense point cloud [1], to which we fit a plane (linear least squares fitting without outlier removal). Over 98% of the roughly 525,000 computed points were less than 0.2 mm away from the computed plane and 64% less than 0.1 mm. The approximate diameter of the reconstructed part of the platter was 80 mm, resulting in a 0.3% relative error in the reconstruction. Refer to figure 4 for the histogram of point-plane distances.

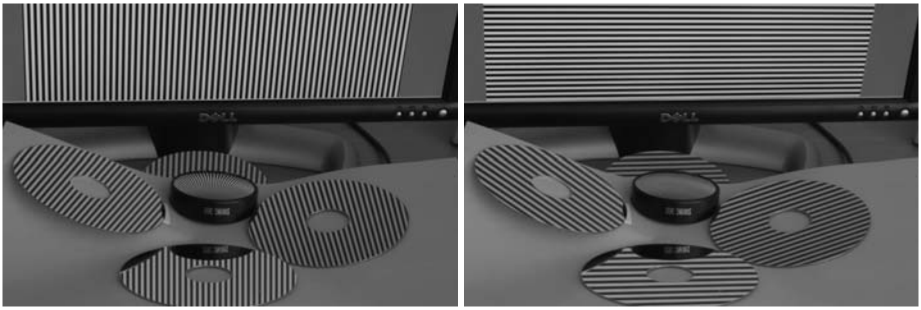


Fig. 2. Two images of our setup. Four planar mirrors (hard disk platters) are placed simultaneously in the scene. The object in the middle is a curved mirror, which was not used for the experiments reported here. The LCD monitor is partly visible in the image, but only its reflections are used to compute camera pose.

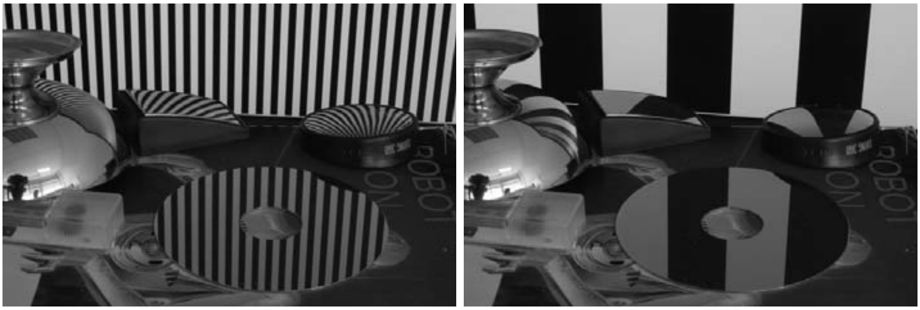


Fig. 3. Two of the images used for the reconstruction of the planar hard drive platter

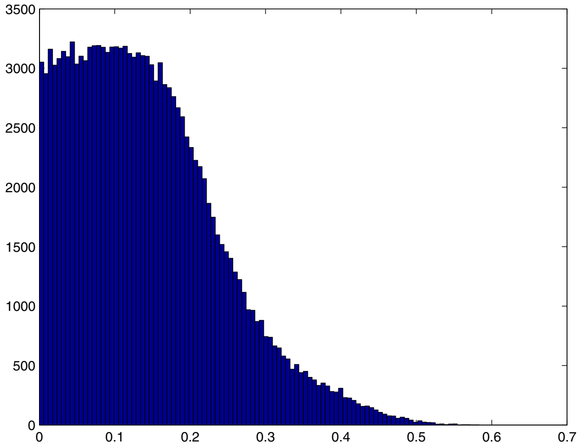


Fig. 4. Point-plane distance. Histogram of the distance of each point to the linear least squares fitted plane (in mm).

9 Conclusions

We have addressed the problem of computing the pose of an object relative to a camera, without any direct view of the object. This problem has to our knowledge not been studied yet. A theoretical study and a practical algorithm have been provided, making use of planar mirrors put in unknown positions in the scene. It was shown that with three mirror positions or more, the problem can in general be solved.

Although rather specific, this problem is very relevant for our work on specular surface reconstruction, which like many similar works uses setups where the camera has not direct view of the reference object.

The method was shown to work with real images, although by an indirect evaluation via a specular surface reconstruction method. A more in-depth evaluation using simulated data should be done, but it seems to be reasonable to assume that the performances will be similar to those of calibration and pose estimation of a camera from several images of a planar calibration grid [13, 9] (the number of parameters and the geometries of the problems are similar).

References

1. T. Bonfort, P. Sturm, P. Gargallo. General Specular Surface Triangulation. ACCV, 2006.
2. Y. Caspi, M. Irani. Alignment of Non-Overlapping Sequences. ICCV, 76-83, 2001.
3. M.A. Halstead, B.A. Barsky, S.A. Klein, R.B. Mandell. Reconstructing Curved Surfaces from Specular Reflection Patterns Using Spline Surface Fitting of Normals. SIGGRAPH, 335-342, 1996.
4. S. Kammel, F. Puente León. Deflectometric measurement of specular surfaces. IEEE Instrumentation and Measurement Technology Conference, 531-536, 2005.

5. S. Maybank. Theory of Reconstruction from Image Motion. Springer Verlag, 1993.
6. M. Oren, S.K. Nayar. A Theory of Specular Surface Geometry. IJCV, 24(2), 1996.
7. S. Savarese, M. Chen, P. Perona. Recovering local shape of a mirror surface from reflection of a regular grid. ECCV, 2004.
8. D. Scharstein, R. Szeliski. High-accuracy stereo depth maps using structured light. CVPR, 195-202, 2003.
9. P. Sturm, S. Maybank. On Plane-Based Camera Calibration. CVPR, 432-437, 1999.
10. P. Sturm. Algorithms for Plane-Based Pose Estimation. CVPR, 706-711, 2000.
11. M. Tarini, H. Lensch, M. Goesele, H.-P. Seidel. 3D acquisition of mirroring objects. Research Report MPI-I-2003-4-001, Max-Planck-Institut für Informatik, 2003.
12. B. Triggs, P.F. McLauchlan, R.I. Hartley, A. Fitzgibbon. Bundle Ajustment – A Modern Synthesis. Vision Algorithms, 298-372, 1999.
13. Z. Zhang. A flexible new technique for camera calibration. PAMI, 22(11), 2000.
14. J.Y. Zheng, A. Murata. Acquiring 3D object models from specular motion using circular lights illumination. ICCV, 1101-1108, 1998.

Dense Motion and Disparity Estimation Via Loopy Belief Propagation

Michael Isard and John MacCormick

Microsoft Research Silicon Valley,
Mountain View, California, USA

Abstract. We describe a method for computing a dense estimate of motion and disparity, given a stereo video sequence containing moving non-rigid objects. In contrast to previous approaches, motion and disparity are estimated simultaneously from a single coherent probabilistic model that correctly accounts for all occlusions, depth discontinuities, and motion discontinuities. The results demonstrate that simultaneous estimation of motion and disparity is superior to estimating either in isolation, and show the promise of the technique for accurate, probabilistically justified, scene analysis.

1 Motivation and Previous Work

The "temporal stereo + motion" problem of estimating the disparity and motion fields in a video sequence of moving objects captured by a calibrated pair of stereo cameras has been studied for at least two decades [1]. It is worthwhile to distinguish between the standard temporal stereo + motion problem, and the more restricted problem of estimating disparity and motion from two consecutive frames in a stereo sequence; we refer to the latter as "two-frame stereo + motion". This paper first introduces a novel solution for two-frame stereo + motion, then explains how to extend the solution to stereo sequences.

Our ultimate objective is to form a reliable, dense 2.5D representation of an image sequence. Acquiring a rectified stereo sequence and running traditional stereo algorithms fills in much of the necessary information, but dense disparity estimation from a single stereo pair is challenging. Matches can be highly ambiguous in non-textured regions; and background regions near foreground object boundaries are only visible in a single camera, meaning their depth must be estimated using only prior information about the shapes of objects in the world.

Exploiting temporal coherence in the stereo sequence can in principle alleviate both of these problems, however as previous work has noted [2], in the absence of explicit motion estimates it is hard to do better than to average out thermal imaging noise in *stationary* regions. We therefore propose to jointly estimate dense motion and disparity in a single coherent probabilistic framework. We show that making use of two-frame motion estimation in conjunction with traditional stereo greatly reduces the regions of the scene which are visible only in a single image. In addition, by filtering over time we are able to propagate information about the depth of scene patches during extended occlusions in the non-reference image.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 32–41, 2006.

Work on temporal stereo+motion has generally been based on sparse image features. This sparsity is not directly compatible with the dense reconstruction of the disparity and motion fields, which is the goal of this paper. Examples of the feature-based approach include [3], which uses line correspondences, and [4].

One significant example that uses optical flow rather than features is [5]. However, this approach employs an iterative segmentation of the scene: an initial estimate is obtained assuming a single rigid motion of the entire scene, then objects with distinct motions are segmented in later iterations by detecting outliers. In contrast, the approach of this paper employs a single probabilistic model from which the motions of all objects are inferred coherently.

Our work is closer in spirit to the large literature on dense stereo reconstruction, including those methods that use belief propagation [6], graph cuts [7], or dynamic programming [8, 9]. However, none of these approaches attempt motion estimation.

Other notable temporal stereo + motion contributions include [10], which achieves excellent accuracy using structured light, and [11, 12], both of which describe interesting algorithms which cannot conveniently be placed in a probabilistic framework.

Our approach to two-frame stereo + motion defines a single Markov random field (MRF) whose nodes are the pixels of the reference image, and whose labels incorporate all possible disparity, motion, and occlusion values. Inference is performed by approximating the MAP estimate for this MRF using loopy belief propagation. As far as we are aware, this is the first work to attempt simultaneous disparity and motion estimation using MRFs. In more abstract terms, however, our approach is distinguished from previous approaches to temporal stereo + motion in three important respects: (i) our estimates are *dense*, in contrast to feature-based approaches such as [3]; (ii) we employ a single *coherent* probabilistic model, in contrast to iterative segmentation approaches such as [5]; and (iii) the likelihoods correctly account for occlusions and discontinuities. We believe this paper presents the first stereo + motion work satisfying all of (i)-(iii).

Item (iii), the modeling of occlusions and discontinuities, can be viewed as a generalization of the occlusion modeling in much previous work on stereo (e.g. [13, 8]). The essential idea is that the likelihood of a particular disparity hypothesis for a particular world point cannot be computed without also specifying whether that point is visible or occluded in each of the images. This "occlusion status" varies in a deterministic fashion near object boundaries. Figure 1 gives a schematic example of this for the stereo + motion problem. One key contribution of this work is that the data likelihoods in the MRF are computed in the following way. The MRF label at a reference image pixel includes an occlusion status (corresponding to the color rendered in figure 1), and this is used in turn to determine which of the non-reference image patches should contribute to the data likelihood. In contrast to much previous work on stereo and motion, patches corresponding to occluded world points are explicitly excluded when they should be.

Our solution to the multi-frame temporal stereo + motion problem amounts to a simple extension of the two-frame MRF. By treating the problem in the

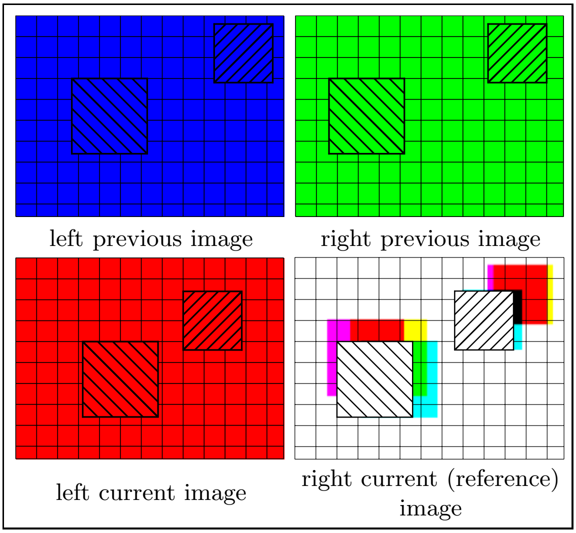


Fig. 1. Motion and disparity determine visibility in non-reference images. Two foreground objects with positive disparities are shown moving against a zero-disparity stationary background. Each pixel in the reference image is colored according to which non-reference images it is visible in. For example, a pixel visible in the left and right previous images but not the left current image is colored blue + green = cyan, pixels visible in all three non-reference images are white, and pixels visible nowhere except the reference image are black.

context of filtering (as opposed to smoothing), the outputs from previous frames can be incorporated by adding an extra term to the MRF data cost.

Section 2 describes the MRF employed for two-frame stereo + motion, and Section 3 explains the extension to the multi-frame case. Section 4 discusses the use of loopy belief propagation to approximate MAP estimates in these MRFs, and Section 5 describes the results.

2 The MRF for Two-Frame Stereo + Motion

The input to the two-frame stereo + motion algorithm consists of four images: Left_0, Right_0, Left_1, Right_1 (which are, respectively, the left and right stereo views of the previous and current frames of a stereo video sequence). The stereo pairs are assumed to be rectified, so that epipolar lines are horizontal, with corresponding pairs occurring on the same scanline.

The output consists, informally, of a complete reconstruction of the disparity and motion fields implied by these four images. To formalize this, we define a graphical model and compute an approximation to the MAP estimate of the disparity and motion fields. The unknowns in the graphical model form a standard four-connected rectangular lattice of the same size as the input images. The nodes are denoted $g_{x,y}$, $x \in \{0, 1, \ldots, X-1\}$, $y \in \{0, 1, \ldots, Y-1\}$, where X, Y

are the width and height, respectively, of the input images. We select the current right-hand image Right_1 to be the reference image, so the state at node $g_{x,y}$, denoted $s_{x,y}$, represents the motion and disparity estimated at pixel (x, y) in Right_1.

The state $s_{x,y}$ at node $g_{x,y}$ models a particular point (or, more realistically, a patch) P on a particular object in the world. P is found by back-projecting a ray from the pixel (x, y) in the reference camera until the ray intersects a scene object. Note that P is fixed on the object, but the object itself may have moved between the previous and current frames. Note also that P may or may not be visible in each of the three non-reference images. The state $s_{x,y}$ is specified by five components. Omitting the x, y suffices, we write $s = (o, d, u, v, w)$, where:

- o is an "occlusion status", described below
- d is P's disparity in the current frame
- u and v are respectively the horizontal and vertical components of P's motion
- w is the difference between P's disparity in the previous frame and the current frame; w can also be thought of as the "depth" component of the motion.

The occlusion status o comprises three binary flags $o = (o_{L1}, o_{L0}, o_{R0})$ specifying whether or not P is visible in the non-reference images. A formal definition of the remaining state variables — d, u, v, w — consists of describing where P projects to in each non-reference image, assuming that it is visible. The definitions adopted are that P projects to

$$\begin{aligned} (x+d, y) &\quad \text{in } \mathsf{Left}_1 \\ (x-u+d-w, y-v) &\quad \text{in } \mathsf{Left}_0 \\ (x-u, y-v) &\quad \text{in } \mathsf{Right}_0. \end{aligned} \tag{1}$$

The posterior probability of the graphical model with states $\{s_{x,y}\}$ is (by definition) the product of some one- and two-node potentials:

$$\mathcal{L} = \prod_{(x,y)} \Phi(s_{x,y}) \prod_{(x,y)\sim(x',y')} \Psi(s_{x,y}, s_{x',y'}), \tag{2}$$

where the second product is over pairs of neighboring nodes.

Maximizing $\mathcal{L}$ is the same as minimizing its negative log, so writing $\phi = -\log \Phi, \psi = -\log \Psi$ we can cast the final objective as minimizing the log posterior: $L = \sum_{(x,y)} \phi(s_{x,y}) + \sum_{(x,y)\sim(x',y')} \psi(s_{x,y}, s_{x',y'})$. The first term here is the *data cost*, discussed next in section 2.1. The second term is the *continuity cost*, discussed in section 2.2.

2.1 Data Cost

The *normalized sum of squares difference* (NSSD) [14] between patches centered at (x, y) in image I and (x', y') in image I' is defined as

$$\mathrm{NSSD}(I, x, y; I', x', y') =$$
$$\frac{\sum_{dx,dy} \|(I_{x+dx,y+dy} - \overline{I}_{x,y}) - (I'_{x'+dx,y'+dy} - \overline{I}'_{x',y'})\|^2}{2\sum_{dx,dy} \left(\|I_{x+dx,y+dy} - \overline{I}_{x,y}\|^2 + \|I'_{x'+dx,y'+dy} - \overline{I}'_{x',y'}\|^2\right)} \tag{3}$$

Here, (dx, dy) ranges over an origin-centered $K \times K$ patch of integers in $\mathbb{Z}^2$; $\| \cdot \|$ is the Euclidean norm in RGB space (i.e. $\mathbb{R}^3$); $I_{x,y}$ is the RGB value (in $\mathbb{R}^3$) of the image I at pixel location (x, y); $\overline{I}_{x,y}$ is the average RGB value of the image I over a $K \times K$ patch centered on (x, y).

Experience has shown that the discriminatory power of the NSSD (3) is improved by changing it in two ways. First, the means $\overline{I}_{x,y}$ are computed with a Gaussian weighting centered on the relevant patch, with a relatively small standard deviation of 0.75 pixels. Second, the NSSD is redefined to be the minimum of (3) over all 2-D sub-pixel shifts of the patch centered at (x, y). The sub-pixel shift can be computed analytically from the image and gradient values within the patch, using the Lucas-Kanade formulas [15].

Obviously, the NSSD is expected to be small for patches derived from different views of the same world point, and arbitrary otherwise. This intuition is captured here by assuming the NSSD is distributed according to some probability law $\Pi(\cdot)$ when the patches correspond, and a distinct probability law $\tilde{\Pi}(\cdot)$ otherwise. The negative log probabilities for these distributions will be written $\pi = -\log \Pi$, $\tilde{\pi} = -\log \tilde{\Pi}$. Numerical values for $\Pi, \tilde{\Pi}$ can be learned from training data or derived from physical assumptions, as described in our technical report [16].

The data cost associated with graph node $g_{x,y}$ in state $s = (o, d, u, v, w)$ can now be defined. First, let

$$\begin{aligned} \mathsf{NSSD}_{L1} &= \mathrm{NSSD}(\mathsf{Right}_1, x, y; \mathsf{Left}_1, x + d, y) \\ \mathsf{NSSD}_{L0} &= \mathrm{NSSD}(\mathsf{Right}_1, x, y; \mathsf{Left}_0, x + d - u - w, y - v) \\ \mathsf{NSSD}_{R0} &= \mathrm{NSSD}(\mathsf{Right}_1, x, y; \mathsf{Right}_0, x - u, y - v) \end{aligned} \tag{4}$$

These definitions have a simple intuitive interpretation. The node $g_{x,y}$ models a world point P. Each of the NSSDs in (4) computes the similarity of two patches that are projections of P: one in the reference image, centered at (x, y), and one in a non-reference image, centered at the location implied by d, u, v, w, as defined by equation (1). However, there is no guarantee that P is actually visible in the non-reference images. In the cases when P is visible, the NSSD will be distributed according to $\Pi(\cdot)$; but when it is occluded, the NSSD is distributed according to $\tilde{\Pi}(\cdot)$. Recalling the definitions of $\pi, \tilde{\pi}$ above, this motivates the definition $\mathsf{Cost}_{L1} = \pi(\mathsf{NSSD}_{L1})$ if $o_{L1} = \mathsf{Visible}$ or $\tilde{\pi}(\mathsf{NSSD}_{L1})$ otherwise, and similarly for Cost_{L0} and Cost_{R0}. These costs are genuine log probabilities, based on the distribution of NSSDs for matched and unmatched patches. Assuming independence between the different NSSD outcomes is equivalent to summing these log probabilities, leading to a total data cost given by $\phi_{x,y}(s) = \mathsf{Cost}_{L1} + \mathsf{Cost}_{L0} + \mathsf{Cost}_{R0}$. Previous work [17] using a similar data cost has shown empirically that the log likelihood ratio of NSSDs, $\pi/\tilde{\pi}$, is well-approximated by a linear function in the region of interest. We take advantage of this here by noting that the above data cost can be expressed in terms of this log likelihood ratio, and adopt a learnt linear function for $\pi/\tilde{\pi}$.

2.2 Continuity Cost

Consider two neighboring nodes g, g' in the graphical model. They are in states $s = (o, d, u, v, w)$ and $s' = (o', d', u', v', w')$ respectively. We would like to derive the continuity cost $\psi(s, s')$. We assume the five components of the state are probabilistically independent, given the image data. Neglecting these dependencies is equivalent to adopting the following functional form for the continuity cost: $\psi(s, s') = \psi_m(o, o') + \psi_d(d, d') + \psi_u(u, u') + \psi_v(v, v') + \psi_w(w, w')$. Reasonable choices for each of these terms can be determined based on expected scene characteristics and the physics of image formation in a calibrated stereo camera rig. For ψ_m, we choose a Potts model with temperature T:

$$\psi_m = \begin{cases} 0 & \text{if } o = o', \\ 1/T & \text{if } o \neq o' \end{cases} \tag{5}$$

where an appropriate value for T can be determined by simulating the Potts model.

For each of the remaining terms in the continuity cost, we assume the absolute difference is distributed such that the negative log of its distribution function has a truncated linear form, for example: $\psi_d(d, d') = \min(a, b\,|d - d'|)$. Our technical report [16] describes how to choose sensible values for a, b based on physical reasoning.

In fact, a need not be constant over the graphical model. Observe that disparity and motion fields are often discontinuous at object boundaries, and object boundaries often occur at locations with high image gradients. This intuition can be incorporated by setting $a = a_0 \exp(-\|\nabla I\|/\alpha)$, where $\|\nabla I\|$ is the gradient of the reference image at the location corresponding to the nodes g, g'. We follow [17] in setting α to be the average value of the image gradient over the whole reference image. However, note that the authors of [17] switch on this so-called "contrast model" only between nodes whose occlusion status differs: this is because [17] deals with 1-D horizontal MRFs, in which a change of occlusion status is guaranteed to correspond to an object boundary. When using 2-D or 3-D MRFs, object boundaries can occur between two neighboring MRF nodes with the same occlusion status. (The simplest example is two vertical neighbors straddling a horizontal object boundary—in this case, both relevant world points are visible in all images.) Hence, our contrast model is switched on for all pairs of neighboring nodes.

3 Temporal Filtering of Stereo + Motion

The previous section described a model for computing disparity and motion fields from two consecutive frames of a stereo video sequence. Clearly, this model could be applied separately to each pair of consecutive frames in a sequence, to obtain disparity and motion fields for the entire sequence. However, we would like to do better: it should be possible to obtain improved estimates by exploiting temporal coherence. This can be achieved with very little extra computational cost, by adopting a *filtering* model in which inferences at time t are influenced by the past — specifically, the output at time $t-1$.

To explain the details of this, some more general notation is needed. Let $\mathcal{G}^{(t)}$ be the MRF for time t, with nodes $g^{(t)}_{x,y}$ and labels $s^{(t)}_{x,y}$. The output of the filtering algorithm at time t is a set of estimated labels $\hat{\mathbf{s}}^{(t)} = \{\hat{s}^{(t)}_{x,y}\}$.

It can be shown [16] that this filtering model is equivalent to adding an extra term to the data cost of Section 2.1, consisting of a *temporal compatibility function* $\gamma(s^{(t)}_{x,y}; \hat{\mathbf{s}}^{(t-1)})$. A plausible form of this temporal compatibility function can be derived as follows. As usual, write the label in terms of its occlusion status, disparity, and motion as $s^{(t)}_{x,y} = (o, d, u, v, w)$, with the occlusion status further broken out into three bits expressing the visibility in the non-reference images: $o = (o_{L,t}, o_{L,t-1}, o_{R,t-1})$. Let P be the world point visible at location (x, y) in the reference image. Then $s_{x,y}$ expresses certain physical facts about P, including the following: if $o_{R,t-1} = \mathsf{Visible}$, then P is visible in image Right_{t-1} at location $x' = x - u, y' = y - v$, with disparity $d' = d - w$. Adopting a constant velocity motion model, we may also assume that P's velocity at time $t-1$ is given by $u' = u$, $v' = v$, $w' = w$.

However, note that the image Right_{t-1} is the reference image for the stereo + motion computation on $\mathcal{G}^{(t-1)}$. Thus (still assuming that $o_{R,t-1} = \mathsf{Visible}$), the MAP estimate for $\mathcal{G}^{(t-1)}$ also has an opinion about P's state: specifically, its opinion is equal to $\hat{s}^{(t-1)}_{x',y'}$, which we write more explicitly as $\hat{s}^{(t-1)}_{x',y'} = (\hat{o}, \hat{d}, \hat{u}, \hat{v}, \hat{w})$.

The temporal compatibility function γ expresses the fact that P's disparity and motion is expected to vary slowly, so this cost should be small when $s_{x,y}$ is close to $\hat{s}_{x',y'}$. A standard choice is to interpret γ as the negative log of a robust distribution function whose components are independent. This is equivalent to taking $\gamma(s_{x,y}; \hat{\mathbf{s}}^{(t)}) = \gamma_d(s_{x,y}, \hat{s}_{x',y'}) + \gamma_u(s_{x,y}, \hat{s}_{x',y'}) + \gamma_v(s_{x,y}, \hat{s}_{x',y'}) + \gamma_w(s_{x,y}, \hat{s}_{x',y'})$, with a robust cost function such as the truncated linear for each component e.g. $\gamma_d(s_{x,y}, \hat{s}_{x',y'}) = \min(a, b\,|d' - \hat{d}|)$ for constants a, b.

However, the previous discussion assumed that point P was visible in Right_{t-1} (i.e. $o_{R,t-1} = \mathsf{Visible}$). If P is not visible, the temporal compatibility function should be uniform. Therefore, the final form adopted for the components of γ is:

$$\gamma_d(s_{x,y}, \hat{s}_{x',y'}) = \begin{cases} \min(a, b\,|d' - \hat{d}|) & \text{if } o_{R,t-1} = \mathsf{Visible}, \\ a & \text{otherwise}, \end{cases}$$

and similarly for $\gamma_u, \gamma_v, \gamma_w$. Our technical report [16] explains how to make sensible choices for a, b.

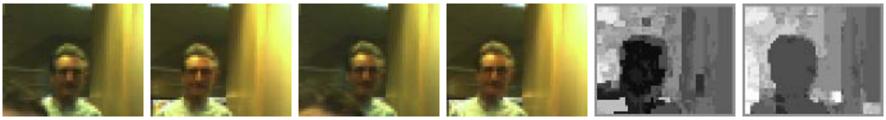
4 Inference for Stereo + Motion

We estimate the MAP of the MRF described in the previous section using the min-sum formulation of loopy belief propagation (BP) [18]. The form of our model allows the use of distance transform techniques [19] which greatly reduce the computational cost, however belief propagation on large images with large disparities and motions remains expensive. It is clear that a multi-resolution approach would help to ameliorate the expense. But note that approaches such as [19], which employ coarser resolutions of the *pixel* (or graph node) space, while retaining the full *state space* resolution, are insufficient: the multiscale algorithm must reduce the number of states considered at each node. We believe it is possible to do this, but the design of such a multiscale algorithm is not at all trivial, and must be postponed to a future paper. Hence, the results presented in the next section employ small, coarsely-subsampled images in order to demonstrate the effect of our stereo + motion algorithm while keeping computational requirements within acceptable limits.

5 Results

We tested our algorithm on several stereo sequences obtained from the public database at `http://www.research.microsoft.com/vision/cambridge/i2i/DSWeb.htm`. The examples shown here are taken from the "Geoff" sequence, focusing on a 100×80 pixel region in the top corner of the sequence, subsampled by a factor of 2 to give 50×40 pixels per frame. For the full stereo + motion computation we use a label space with maximum values of $|o| = 8$, $|d| = 8$, $|u| = 8$, $|v| = 3$, $|w| = 1$, giving 1536 labels per node. The small image size and restricted range of disparity and motion are chosen for computational convenience, however the power of the approach is demonstrated even on this limited example.

Figure 2 demonstrates resistance to fast-moving occluders. When a nearby foreground object moves in from the left the stereo computation alone is unable to accurately estimate the foreground disparity in the newly-occluded region. The filtered stereo + motion algorithm correctly uses information from previous timesteps to recover a reasonable disparity estimate. The 2-frame stereo+motion algorithm, not shown, has a slightly noisier output but avoids the gross artifact.



(a) left previous image

(b) right previous image

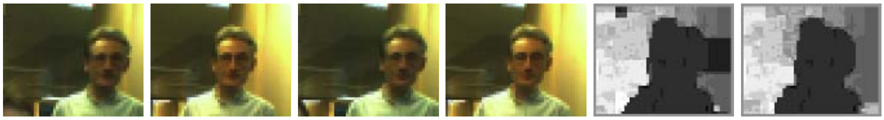
(c) left current image

(d) right current image

(e) disparity estimated from stereo alone

(f) disparity estimated from filtered stereo + motion

Fig. 2. Stereo+motion estimates disparity through transient occlusions. An occluder has appeared in the bottom corner of the left current image (c) but not yet in the right (d). The stereo computation alone (e) does not have enough information to estimate the disparity in this region, but the filtered stereo+motion algorithm (f) uses information from previous timesteps to improve the result.



(a) left previous image

(b) right previous image

(c) left current image

(d) right current image

(e) disparity estimated from two-frame stereo + motion

(f) disparity estimated from filtered stereo + motion

Fig. 3. Stereo+motion propagates disparity estimates through multiple frames. The foreground person has stopped moving, and there is a large left occlusion in the textureless area on the right hand side of the image. The two-frame stereo computation (e) has no information about the disparities in this occluded region and the lack of texture causes a large artifact. The filtered stereo+motion estimate (f) propagates disparity estimates from previous frames to stabilise the difficult region.

Figure 3 shows an additional benefit of temporal filtering. The right hand edge of the image is textureless and the foreground person is almost stationary, hence neither the disparity alone nor two-frame stereo + motion can accurately estimate the disparity where the wall is occluded in the left image. Since the foreground person was previously further to the left, there was a reliable disparity estimate on the wall at an earlier frame, and the filtering algorithm has propagated this estimate in the absence of new information.

The full filtering algorithm for the examples shown takes around 5 s per frame in a C++ implementation running on a 2.2GHz Intel Xeon workstation. For comparison, the disparity-only computation on this small image patch takes 330 ms per frame; comparing with the state of the art suggests there is substantial room for improvement if performance were critical.

6 Conclusions

An algorithm was presented to solve the temporal stereo + motion problem. We believe this is the first such algorithm to obtain dense disparity and motion estimates using a coherent probabilistic framework with physically correct occlusion labels. The approach models a two-frame stereo + motion problem as a single MRF, and extends to the multi-frame case by using temporal filtering in the same MRF framework.

The results confirm that dense stereo + motion produces superior results to stereo alone. The estimates for both stationary and moving objects are stabilized, exhibiting less flicker. Additionally, there are certain image regions in which stereo alone has no information, but stereo + motion does have information in (the majority of) those regions, and can therefore infer correct disparity and motion fields there.

The clearest opportunity for future work is in decreasing the computational expense of the algorithm, and the most obvious avenue for this is a multi-scale approach. This is presently an object of active research.

References

1. Waxman, A., Duncan, J.: Binocular image flows: Steps towards stereo-motion fusion. IEEE Trans. on PAMI **8** (1986) 715–729
2. Williams, O., Isard, M., MacCormick, J.: Estimating disparity and occlusions in stereo video sequences. In: Proc. CVPR. (2005)
3. Chang, Y., Aggarwal, J.: Line correspondences from cooperating spatial and temporal grouping processes for a sequence of images. Computer Vision and Image Understanding **67** (1997) 186–201
4. Ho, A., Pong, T.: Cooperative fusion of stereo and motion. Pattern Recognition **29** (1996) 121–130
5. Wang, W., Duncan, J.: Recovering the three dimensional motion and structure of multiple moving objects from binocular image flows. Computer Vision and Image Understanding **63** (1996) 430–446
6. Sun, J., Shum, H.Y., Zheng, N.N.: Stereo matching using belief propagation. In: Proc. European Conf. on Computer Vision. (2002) 510–524
7. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: Proc. Int. Conf. on Computer Vision. (2001)
8. Criminisi, A., Shotton, J., Blake, A., Torr, P.: Gaze manipulation for one-to-one teleconferencing. In: Proc. Int. Conf. on Computer Vision. (2003)
9. Leung, C., Appleton, B., Lovell, B.C., Sun, C.: An energy minimisation approach to stereo-temporal dense reconstruction. In: Proc. Int. Conf. on Pattern Recognition. (2004)
10. Zhang, L., Curless, B., Seitz, S.M.: Spacetime stereo: Shape recovery for dynamic scenes. In: Proc. CVPR. Volume 2. (2003) 367–374
11. Shao, J.: Generation of temporally consistent multiple virtual camera views from stereoscopic image sequences. Int. J. Comput. Vision **47** (2002) 171–180
12. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. In: Proc. Int. Conf. on Computer Vision. Volume 2. (1999) 722–729
13. Belhumeur, P.: A Bayesian approach to binocular stereopsis. Int. J. Computer Vision **19** (1996) 237–260
14. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Computer Vision (2002)
15. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. Int. J. Comput. Vision **56** (2004) 221–255
16. Isard, M., MacCormick, J.: Dense motion and disparity estimation via loopy belief propagation. Technical report, Microsoft Research (2005)
17. Blake, A., et al.: Bi-layer segmentation of binocular stereo video. In: Proc. CVPR. (2005)
18. Yedidia, J., Freeman, W., Weiss, Y.: Understanding Belief Propagation and its Generalizations. In: Exploring Artificial Intelligence in the New Millennium. Elsevier Science (2003)
19. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. In: Proc. CVPR. (2004)

A Real-Time Large Disparity Range Stereo-System Using FPGAs

Divyang K. Masrani and W. James MacLean

Department of Electrical and Computer Engineering, University of Toronto,
Toronto, Ontario, Canada
{masrani, maclean}@eecg.toronto.edu

Abstract. In this paper, we discuss the design and implementation of a Field-Programmable Gate Array (FPGA) based stereo depth measurement system that is capable of handling a very large disparity range. The system performs rectification of the input video stream and a left-right consistency check to improve the accuracy of the results and generates subpixel disparities at 30 frames/second on 480×640 images. The system is based on the Local Weighted Phase-Correlation algorithm [9] which estimates disparity using a multi-scale and multi-orientation approach. Though FPGAs are ideal devices to exploit the inherent parallelism in many computer vision algorithms, they have a finite resource capacity which poses a challenge when adapting a system to deal with large image sizes or disparity ranges. In this work, we take advantage of the temporal information available in a video sequence to design a novel architecture for the correlation unit to achieve correlation over a large range while keeping the resource utilisation very low as compared to a naive approach of designing a correlation unit in hardware.

1 Introduction

Stereo disparity estimation is a prime application for embedded computer vision systems. Since stereo can provide depth information, it has potential uses in navigation systems, robotics, object recognition and surveillance systems, just to name a few. Due to the computational complexity of many stereo algorithms, a number of attempts have been made to implement such systems using hardware [2, 10, 14, 19], including reconfigurable hardware in the form of FPGAs [6, 11, 20, 12, 18, 5]. In related work, [1] implements Lucas & Kanade optic flow using FPGAs. Solutions based on reconfigurable hardware have the desirable property of allowing the designer to take advantage of the parallelism inherent in many computer vision problems, not the least of which is stereo disparity estimation.

While designing with FPGAs is faster than designing Application Specific ICs (ASICs), it suffers from the problem of fixed resources. In an application based on a serial CPU or DSP, one can typically add memory or disk space to allow the algorithm to handle a larger version of the same problem, for example larger image sizes or increased disparity ranges in the case of stereo. System performance may suffer, but the new system still runs. In the case of FPGA-based systems, there is a finite amount of logic available, and when this is exhausted the only solution is to add another device

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 42–51, 2006.

or modify the algorithm. Not only is this costly from the design point of view, but may also involve the additional design issue of how to partition the logic across several devices.

In this paper we present the development of a versatile real-time stereo-vision platform. The system is an improvement of an earlier one [5] and addresses specific limitations of the previous system; capability to handle very large disparities, improving the accuracy of the system by pre-processing (input image rectification) and post-processing (consistency check), and finally the ability to handle larger images. The highlight of the work is the development of a novel architecture for the *Phase Correlation Unit* that can handle the correspondence task for scenes with very large disparities, but without increased resource usage on the FPGA, as compared to [5] which is capable of handling a disparity of only 20 pixels. The key to achieving large disparity correspondence matches is the use of a shiftable correlation window that tracks the disparity estimate for each pixel over time, as well as a roving correlation window that explores the correlation surface outside the range of the tracking window in order to detect new matches when the shiftable window is centred on an incorrect match. The basic assumption is that, in most cases, disparity values do not change radically between frames, thus allowing some of the computation to be spread over time.

In Section 2, we briefly outline the technology used in this work and the platform used for the system development. In Section 3, we cover the theoretical basis of the phase-based stereo algorithm and then describe the architecture and implementation of the system. Section 4 discusses the results and the use of the *correlation unit* in alternate situations.

1.1 Previous Work

A variety of reconfigurable stereo machines have been reported [18, 12, 20, 6, 11]. The PARTS reconfigurable computer [18] consists of a 4 × 4 array of mesh-connected FPGAs with a maximum total number of about 35,000 4-input LUTs. A stereo system was developed on PARTS hardware using the census transform, which mainly consists of bit-wise comparisons and additions [20]. Kanade *et al.*[12] describe a hybrid system using C40 digital signal processors together with programmable logic devices (PLDs, similar to FPGAs) mounted on boards in a VME-bus backplane. The system, which the authors do not claim to be reconfigurable, implements a sum-of-absolute-differences along predetermined epipolar geometry to generate 5-bit disparity estimates at frame-rate. In Faugeras *et al.*[6], a 4 × 4 matrix of small FPGAs is used to perform the cross-correlation of two 256×256 images in 140 ms. In Hou *et al.*[11], a combination of FPGA and Digital Signal Processors (DSPs) is used to perform edge-based stereo vision. Their approach uses FPGAs to perform low level tasks like edge detection and uses DSPs for higher level integration tasks. In [5] a development system based on four Xilinx XCV2000E devices is used to implement a dense, multi-scale, multi-orientation, phase-correlation based stereo system that runs at 30 frames/second (fps). It is worth noting that not all previous hardware approaches have been based on reconfigurable devices. In [13], a DSP-based stereo system performing rectification and area correlation, called the SRI Small Vision Module, is described. ASIC-based designs are reported in [16, 2] and in [19] commodity graphics hardware is used.

2 Reconfigurable Computing Platform

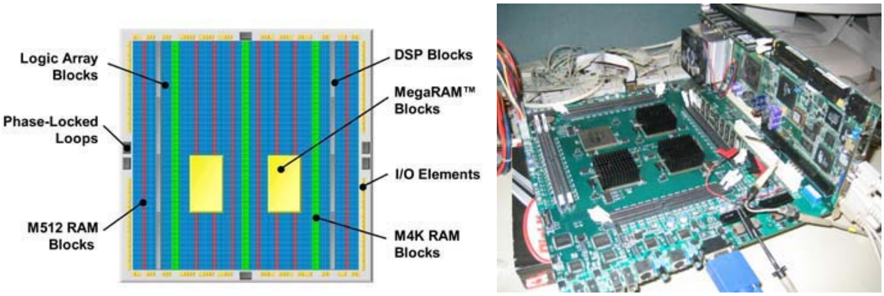
2.1 Field-Programmable Gate Arrays

An FPGA is an array of logic gates whose behaviour can be programmed by the end-user to perform a wide variety of logical functions, and which can be reconfigured as requirements change. FPGAs generally consist of four major components: 1) Logic blocks/elements (LB/LE); 2) I/O blocks; 3) Logic interconnect; and 4) dedicated hardware circuitry. The logic blocks of an FPGA can be configured to implement basic combinatorial logic (AND, OR, NOR, etc gates) or more complex sequential logic functions such as as microprocessor. The logic interconnect in an FPGA consists of wire segments of varying lengths which can be interconnected via electrically programmable switches. The density of logic blocks used in an FPGA depends on the length and number of wire segments used for routing.

Most modern FPGAs also have various dedicated circuitry in addition to the programmable logic. These come in the form of high-speed and high-bandwidth embedded memory, dedicated DSP blocks, Phase-Locked Loops (PLLs) for generating multiple clocks, and even general purpose processors. The FPGA we are using in our system, the Altera Stratix S80, comes with three different memory block sizes; 512 bits, 4 Kbits, and 512 Kbits for a maximum of 7 Mbits of embedded memory and 22 DSP blocks consisting of multipliers, adders, subtractors, accumulators, and pipeline registers. Figure 1 (a) shows an overview of the Altera Stratix S80 chip [3].

2.2 Transmogrifier-4Reconfigurable Platform

The TransmogrifierFour [7] (b) is a general purpose reconfigurable prototyping system containing four Altera Stratix S80 FPGAs. The board has specific features to support image processing and computational vision algorithms; these include dual-channel NTSC and FireWire camera interfaces, video encoder/decoder chip, and 2GB of DDR RAM connected to each FPGA. Each FPGA is also connected to the other three FPGAs and a PCI interface is provided to communicate with the board over a network. This can be used to send control signals or for debugging. The board with its major components is shown in Figure 1 (b).



(a) (b)

Fig. 1. (a)Typical features of a modern FPGA [3]. (b) Transmogrifier-4 reconfigurable computing board [7].

3 Large Disparity Stereo-System Development

The system implemented in this work is based on the "Local Weighted Phase Correlation" (LWPC) algorithm [9], which estimates disparity at a set of pre-shifts using a multi-scale, multi-orientation approach. A version of this algorithm was implemented in [5] but the system is limited to handling a maximum disparity of 20 pixels due to resource limitations on the FPGA. In the current implementation, we use two shiftable windows in the correlation unit to increase the disparity range of the system to 128 pixels (theoretically, the system can be implemented to handle a disparity range as large as the image width) without an increase in resource usage. There is a trade-off between the maximum disparity the system can handle and the time to initialise the system or recover from a mismatch, typically in the range of few tens of milliseconds.

3.1 Temporal Local-Weighted Phase Correlation

Based on the assumption that at video-rate (30 fps) the disparity of a given pixel will not change drastically from one frame the next, we use temporal information by performing localised correlation using a window centred on the disparity a pixel is expected to have at the current frame. This is discussed further below where we describe the architecture of the *Phase Correlation Unit*. Disparity calculations are performed at three scales$(1, 2, 4)$ and in three orientations (-45^o, 0^o, $+45^o$), the results of which are summed across scale and orientation. The expected interval between false peaks is approximately the wavelength of the filters applied at each scale. Thus the false peaks at different scales occur at different disparities and summation over the scales yields a prominent peak only at the true disparity [9]. The details of the LWPC algorithm can be found in [8]. Step 2 of the algorithm reflecting the incorporation of the temporal information is shown below:

2. For each scale and orientation, compute local voting functions $C_{j,s}(x,\tau)$ in a window centred at τ_c as

$$C_{j,s}(x,\tau) = \frac{W(x) \otimes [O_l(x) O_r^*(x+\tau)]}{\sqrt{W(x) \otimes |O_l(x)|^2} \sqrt{W(x) \otimes |O_r(x)|^2}} , \tag{1}$$

where $W(x)$ is a smoothing, localized window and τ is the pre-shift of the right filter output centred at the disparity of the pixel from the previous frame.

In addition, pre-processing (image rectification) and post-processing (left-right / right-left validation check) stages are also implemented to increase the accuracy of the system.

3.2 System Architecture

The high level architecture of the complete system is shown in Figure 2. It consists of six major units: Video Interface unit, Image Rectification unit, Scale-Orientation Decomposition unit, Phase-Correlation unit, Interpolation and Peak Detection unit, and Consistency Check unit.

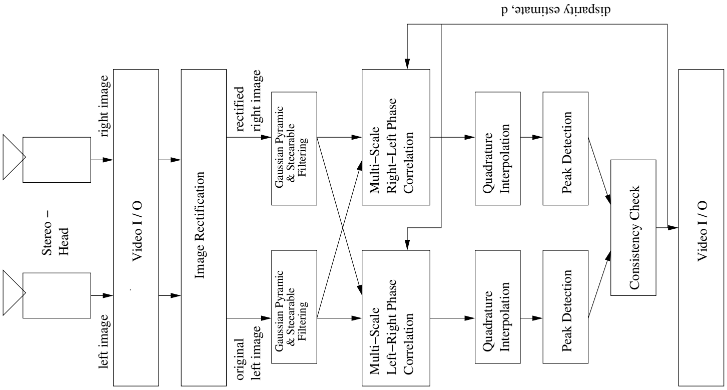


Fig. 2. High-level architecture of the stereo system

The **Video Interface Unit** is capable of receiving video signals from either NTSC or FireWire cameras at 30 fps and an image size of 480×640. In addition to the pixel values, the *Video Interface Unit* output "new line" and "new frame"signals. The data is sent to the *Image Rectification Unit* as it arrives without any buffering. This unit runs on the *camera clock*.

The **Image Rectification Unit** (Figure 3) treats the left input as the reference image and rectifies the right input using bilinear interpolation [17]. A stereo-setup with a worst-case vertical misalignment of 32 *scanlines* between the left and right image is assumed, which requires buffering of 64 *scanlines* of both the left and right image. This unit, as the rest of the system except the *Video I/O Unit*, run on the *system clock*. A *synchroniser circuit* is designed to handle glitch-free transfer of data between the two asynchronous clocks.

The warping operation for image rectification is approximated using the following bicubic polynomial:

$$\begin{aligned} x^{'} &= a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 \\ &\quad + a_6 x^3 + a_7 x^2 y + a_8 xy^2 + a_9 y^3 \\ y^{'} &= b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy + b_5 y^2 \\ &\quad + b_6 x^3 + b_7 x^2 y + b_8 xy^2 + b_9 y^3 \ , \end{aligned} \tag{2}$$

where the a_i and b_i coefficients are computed by offline calibration.

The **Scale-Orientation Decomposition Unit** first builds a three-level Gaussian Pyramid by passing the the incoming right and left images through low-pass filters and sub-sampling. The pyramids are then decomposed into three orientations (-45^o, 0^o, +45^o) using G2/H2 steerable filters. G2/H2 filtering is implemented using a set of seven basis filters. By choosing a set of proper coefficients for the linear combination of the basis filters, filters of any arbitrary orientation can be synthesised. Since G2/H2 filters are X-Y separable, they require considerably less hardware resources than non-separable filters. The filter output is reduced to a 16-bit representation which is then sent to the Phase-Correlation unit.

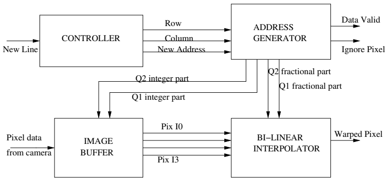


Fig. 3. Architecture of Image Rectification Unit

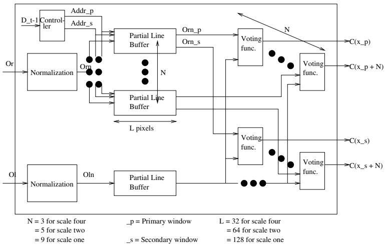


Fig. 4. Modified correlation unit with two shiftable windows

The **Phase-Correlation Unit** computes the real part of the voting function $C_{j,s}(x, \tau)$ as mentioned in Eq. 1 for all $1 \leq s \leq S$, $1 \leq j \leq F$, $0 \leq \tau \leq D$, where S is the total number of scales, F is the total number of orientations, and D is the maximum allowed disparity.

The *Phase Correlation Unit* is implemented using two shiftable correlation windows (see Figure 4) instead of a fixed window as is the traditional approach. One window, the *Primary Tracking Window* (*PTW*) uses temporal information to perform correlation in a localised region for each pixel. The tracking algorithm is currently a very simple one; the window is centred at the disparity estimate from the previous frame for a given pixel. More complex algorithms can be used as discussed in Section 4. When propagating disparity estimates between frames, it is necessary to consider that such algorithms suffer from the risk of getting stuck in a local minima (wrong matches) [4], especially during the initial frames. We have employed an initialisation stage to obtain an accurate disparity map. A second window, the *Secondary Roving Window* (*SRW*) (see Figure 5) does an incremental search up to a user-specifiable maximum disparity value. The increments are set equal to length of the correlation window, L, but these can be modified by a user at *run-time*. The *SRW* also aides in recovery from a mismatch after the initialisation stage. In situations where a new object enters the scene or a region becomes dis-occluded, the *SRW* will pick up this new information and provide a disparity estimate with a higher confidence value than the *PTW*, which can then latch on to this new estimate. There is a tradeoff between the time to recovery from a mismatch and the maximum disparity that the system can handle. For a maximum disparity of 128 pixels with increments of 10 pixels per frame for the *SRW*, the worst-case time to recovery is 233 milliseconds.

The **Interpolation/Peak-Detection Unit** interpolates the voting function results, $C_{j,2}(x, \tau)$ and $C_{j,4}(x, \tau)$, from the two coarser scales, in both x and τ domains such that they can be combined with the results from the finest scale, $C_{j,1}(x, \tau)$. Quadrature interpolation is performed in the τ domain and constant interpolation in the x domain. The interpolated voting functions are then combined across the scales and orientations to produce overall voting function $C(x, \tau)$. The peak in the voting function is then detected for each pixel as the maximum value of $C(x, \tau)$.

The **Consistency Check Unit** receives the estimated disparity results from both left-right and right-left correlations and performs a validity check on the results. The disparity value is accepted as valid if the results from the two correlation windows do not

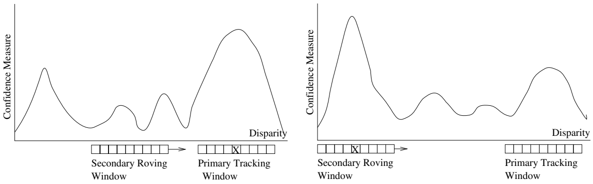


Fig. 5. PTW is correctly tracking the peak (denoted by an **X**) in the confidence measure in (a). In (b), PTW has lost track of the peak, but SRW has picked it up. PTW will latch on to this estimate at the next frame.

differ by more than one pixel. The checked disparity values are then sent back to the video interface unit to be displayed on a monitor. The invalid values are assigned special flag for display purposes.

4 Performance and Suggestions

The stereo system presented in this paper performs multi-scale, multi-orientation depth extraction for disparities up to 128 pixels using roughly the same amount of hardware resource as the previous system that is capable of handling disparities of only 20 pixels [5]. A dense disparity map is produced at the rate of 30 frames / second for an image size of 480 x 640 pixels. In terms of the Points x Disparity per second metric measure, the system is theoretically capable of achieving a performance of over 330 million PDS, which is considerably greater than the any of the others listed [18, 5].

To better understand the workings of the modified correlation unit, we look at results from two real image sequences. The first, MDR-1, is a scene with a static camera and a moving person, and has a maximum disparity of around 16 pixels. The second, MDR-2, is a more complex scene with a moving person and a moving camera, and has a maximum disparity of approximately 30 pixels.

Frame 2 of the MDR-1 sequence is shown in Figure 6 (a). The disparity map during the initialisation stage is shown in (Figure 6 (c)) and the disparity map once the system has settled into the global minimum is shown in Figure 6 (d). For this particular sequence the algorithm settles into the global minimum by the second frame. The disparity map from the fixed correlation window of [5] is shown in Figure 6 (b) for comparison.

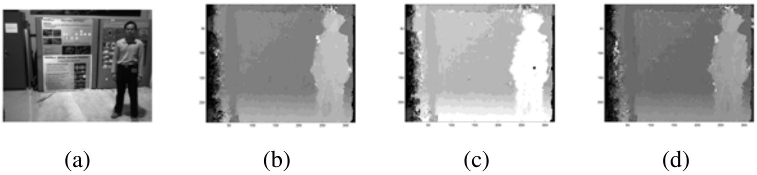

(a) (b) (c) (d)

Fig. 6. In sequence MDR-1, we see that the proposed range-expansion algorithm (d) matches the original algorithm (b) by frame 2. The first frame from the range-expansion algorithm is shown in (c).

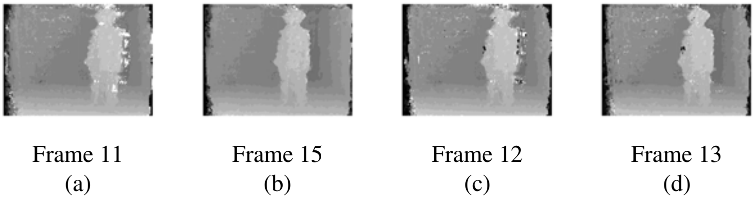


Fig. 7. The recovery time for the system with a maximum secondary shift of 70 pixels is shown in (a) and (b). This can be reduced by using a smaller maximum shift, *e.g.* 30 pixels as shown in (c) and (d). In the latter case, recovery occurs in one frame as opposed to four.

In Figure 7 we show the difference in recovery time for the cases when the secondary correlation window is shifted up to a disparity of: i) 70 pixels and ii) 30 pixels. Figure 7 (a) shows frame 11 for case (i); the results start to deteriorate but are completely recovered by frame 15, Figure 7 (b). For case (ii), the results deteriorate at frame 12, Figure 7 (c), and are already recovered by frame 13, Figure 7 (d). In the MDR-1 sequence, we know that the maximum disparity is around 16 pixels and in such cases where we have prior knowledge of the scene, the ability to select the maximum disparity parameter can yield better results. The disparity maps from the MDR-2 sequence for frame 4 (Figure 8 (a)) are shown in Figure 8 (b) for the implementation in [5] and Figure 8 (c) for our implementation. In [5], where the maximum disparity is limited to 20 pixels, the system cannot handle this sequence whereas our system shows good results.

A number of variations of the design can be implemented to achieve better results without having to make any changes to the correlation unit. Instead of the simple tracker that we are currently using for the *PTW*, a tracker based on a constant-velocity motion model can be used to achieve better tracking. The velocity estimate can be obtained by taking the difference between disparities in the previous two frames, $v_t = d_{t-2} - d_{t-1}$, where v_t is the predicted disparity velocity for the current frame. Similarly, the location of the secondary window can be computed using a probabilistic likelihood estimate instead of the pre-determined roving locations. Other options include the possibility of concatenating the two correlation windows after the initialisation stage so as to support greater movement of objects from one frame to the next. The decision of when to concatenate the windows and when to use them individually in parallel can be made by

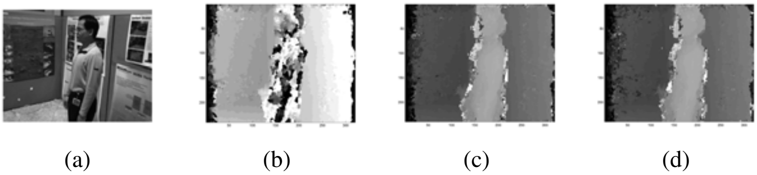


Fig. 8. In sequence MDR-2, we see that the proposed range-expansion algorithm (c) performs significantly better than the original algorithm (b). The disparity map using a larger primary correlation window of 13 pixels (d) is a slight improvement over (c).

a simple count of the number of invalid disparity estimates after the validation check phase. This can be done for the whole image, region by region, or even for individual pixels. The issue of boundary overreach in correlation based algorithms [15] can also be solved by simply shifting the correlation windows by $\pm L/2$, where L is the length of the correlation window, so that the window does not cross over an object boundary. All of these modifications require the implementation of a post-processing stage that generates the appropriate input parameters for the correlation unit without having to make internal changes to the correlation unit itself.

The use of the correlation unit is not limited to a stereo-system. It can also be used in other systems such as object recognition using template matching, for *e.g.*, appearance models for object recognition. The two correlation windows can be used independently to search different regions of an image thereby speeding up the search process or they can be combined to support a larger template.

5 Summary

We have presented an FPGA-based real-time stereo system that is capable of handling very large disparities using limited hardware resources. We achieve this by designing a novel architecture for the correlation unit and also suggest possible uses of the correlation unit in variations of the stereo algorithm and even uses in different algorithms.

References

1. Javier Díaz Alonso. Real-time optical flow computation using FPGAs. In *Proceedings of the Early Cognitive Vision Workshop*, Isle of Skye, Scotland, June 2004.
2. Peter J. Burt. A pyramid-based front-end processor for dynamic vision applications. *Proceedings of the IEEE*, 90(7):1188–1200, July 2002.
3. Altera Corporation. Stratix devices. http://www.altera.com/products/devices/stratix/stx-index.jsp, 2003.
4. S. Crossley, N. A. Thacker, and N. L. Seed. Robust stereo via temporal consistency. In *Proceedings of the British Machine Vision Conference*, pages 659–668, 1997.
5. Ahmad Darabiha, Jonathan Rose, and W. James MacLean. Video-rate stereo depth measurement on programmable hardware. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, volume 1, pages 203–210, Madison, WI, June 2003.
6. Olivier Faugeras, Bernard Hotz, Hervé Mathieu, Thierry Viéville, Zhengyou Zhang, Pascal Fua, Eric Théron, Laurent Moll, Gérard Berry, Jean Vuillemin, Patrice Bertin, and Catherine Proy. Real time correlation-based stereo: Algorithm, implementations and applications. Technical Report Research Report 2013, INRIA Sophia Antipolis, August 1993.
7. Josh Fender. Transmogrifier 4 preliminary information. http://www.eecg.toronto.edu/˜fender/tm4/sointroduction.shtml, August 2003.
8. David J. Fleet. *Measurement of Image Velocity*. Kluwer Academic Publishers, Boston, Massachusetts, 1992.
9. David J. Fleet. Disparity from local weighted phase correlation. In *International Conference on Systems, Man and Cybernetics*, volume 1, pages 48–54, 1994.
10. Heiko Hirschmüller, Peter R. Innocent, and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1/2/3):229–246, 2002. stereo,intensity correlation,MMX, fast.

11. K. M. Hou and A. Belloum. A reconfigurable and flexible parallel 3d vision system for a mobile robot. In *IEEE Workshop on Computer Architecture for Machine Perception*, New Orleans, Louisiana, December 1993.
12. Takeo Kanade, Atsushi Yoshida, Kazuo Oda, Hiroshi Kano, and Masaya Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *Proceedings of the 15th IEEE Computer Vision & Pattern Recognition Conference*, pages 196–202, San Francisco, June 1996.
13. Kurt Konolige. Small vision systems: Hardware and implmentation. In *Proceedings of the Eighth International Symposium on Robotics Research (Robotics Research 8)*, pages 203–212, Hayama, Japan, October 1997.
14. Karsten Mühlmann, Dennis Maier, Jürgen Hesser, and Reinhard M. Anner. Calculating dense disparity maps from color stereo images, an efficient implementation. *International Journal of Computer Vision*, 47(1/2/3):79–88, 2002. stereo,intensity correlation,MMX,fast.
15. M. Okutomi and Y. Katayama. A simple stereo algorithm to recover precise object boundaries and smooth surfaces. In *Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision—SMBV'01*, 2001.
16. G. van der Wal and P. Burt. A VLSI pyramid chip for multiresolution image analysis. *Int. Journal of Computer Vision*, 8:177–190, 1992.
17. George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1994.
18. J. Woodfill and B. Von Herzen. Real time stereo vision on the parts reconfigurable computer. In *5th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 201–210, 1997.
19. R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *Proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition*, pages 211–218, Madison, Wisconsin, June 2003.
20. R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the 3rd European Conference on Computer Vision*, pages 150–158, May 1994. http://www.cs.cornell.edu/rdz/Papers/Archive/neccv.ps, http://www.cs.cornell.edu/rdz/Papers/Archive/nplt-journal.ps.gz.

Use of a Dense Surface Point Distribution Model in a Three-Stage Anatomical Shape Reconstruction from Sparse Information for Computer Assisted Orthopaedic Surgery: A Preliminary Study

Guoyan Zheng, Kumar T. Rajamani, and Lutz-Peter Nolte

MEM Research Center, University of Bern, Stauffacherstrasse 78, CH-3014, Switzerland
Guoyan.Zheng@MEMcenter.unibe.ch

Abstract. Constructing anatomical shape from extremely sparse information is a challenging task. *A priori* information is often required to handle this otherwise ill-posed problem. In the present paper, we try to solve the problem in an accurate and robust way. At the heart of our approach lies the combination of a three-stage anatomical shape reconstruction technique and a dense surface point distribution model (DS-PDM). The DS-PDM is constructed from an already-aligned sparse training shape set using Loop subdivision. Its application facilitates the setup of point correspondences for all three stages of surface reconstruction due to its dense description. The proposed approach is especially useful for accurate and stable surface reconstruction from sparse information when only a small number of *a priori* training shapes are available. It adapts gradually to use more information derived from the *a priori* model when larger number of training data are available. The proposed approach has been successfully validated in a preliminary study on anatomical shape reconstruction of two femoral heads using only dozens of sparse points, yielding promising results.

1 Introduction

With the recent introduction of navigation techniques in orthopedic surgery, three dimensional (3D) models of the patient are routinely used to provide image guidance and enhanced visualization to a surgeon to assist in navigation and planning. An obvious way to derive a 3D model is to extract it from tomographic data, i.e. images obtained from Computed Tomography (CT) or Magnetic Resonance Imaging (MRI). To avoid the high costs and possible health hazards (CT-imaging) associated with such scans, an alternative way is to reconstruct surface using partial data consisting of landmarks and surface points which are interactively digitized by the surgeon or obtained from intra-operative imaging means such as ultrasound [1] or fluoroscopy [2].

Constructing a patient-specific 3D model from extremely sparse data is a challenging task. Additionally, inherent to the navigation application is the high accuracy requirement. When surface reconstruction is used for the purpose of surgical guidance, a target error of less than 1.5 mm on average is normally required [3]. An effective technique to solve this problem to incorporate the *a priori* information of the desired objects into the reconstruction process [4].

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 52 – 60, 2006.

One way to incorporate the *a priori* information is to use the Point Distribution Model (PDM) by Cootes et al. [5], which can learn shape variations from a set of training data, containing a set of landmarks to define the shape. When applied to surface reconstruction, this approach employs Principal Component Analysis (PCA) to reduce the dimensionality of the shape parameter space and then performs shape prediction in the reduced, low dimensional space based on the intra-operative measurements. Using such an approach for shape prediction is essentially equivalent to assuming that future instances are generated by a Gaussian distribution, and that its parameters – mean and covariance – could be exactly estimated from the training data [6]. In the generative case, the Law of Large Numbers justifies using this method as long as the number of samples are big enough. However, it may well be that the measurement can not be fully accounted for by any element generated from this distribution due to [7]: (1) our measurements may be deteriorated due to noise or other sources of errors; (2) we cannot expect to cover the full range of the object class with limited number of training samples. Therefore, in the seminal work of Blanz and Veter for the synthesis of 3D faces using a morphable model [8], Mahalanobis distance was employed to achieve a tradeoff between fidelity and plausibility.

Another way to incorporate the *a priori* information is shape deformation [9]. Starting from a template shape and a sparse set of paired points established between the digitized points and their homologous correspondences on the template shape, the surface is reconstructed by warping the template shape to fit the digitized points. The template shape and the smoothness requirement of the warping could be regarded as another way to incorporate the *a priori* information. Theoretically it is possible to reconstruct any homologous surface by this method if enough homologous corresponding point pairs are given. However, the reconstruction quality depends on how closely the template shape is similar to the target shape, when only a sparse set of homologous corresponding point pairs are given.

Within the scope of computer assisted orthopedic surgery, surface reconstruction from partial data starts with the seminal work of Fleute et al. [10]. In their work, octree splines are used to align and match the training shapes and then the statistical shape model is fitted to the sparse intra-operative data via jointly optimizing morphing and pose. In a recent application of their algorithm to Total Knee Arthroplasty (TKA), as many as 500 points are required [11]. Chan et al [12] use a similar algorithm, but optimize morphing and pose separately using an iterative closest point (ICP) method. No regularization is used in both methods. Following the seminal work of Blanz and Veter [7], our prior work [13-15] focuses on developing robust and stable approach for anatomical shape reconstruction. Mahalanobis distance was also employed to achieve a stable solution [13]. It can be relaxed when more and more points are incorporated [14]. Recently, outlier rejecting mechanism has been incorporated based on robust statistics [15]. However, due to the small number of training shapes, the accuracy in our prior work can not satisfy the requirement of surgical guidance and the target application is then only for enhanced 3D visualization.

In this paper, we try to solve the problem in an accurate and robust way. At the heart of our approach lies the combination of a three-stage anatomical shape reconstruction technique and an *a priori* dense surface point distribution model (DS-PDM). The *a priori* DS-PDM is constructed from an already-aligned sparse training shape set using Loop subdivision. The reconstruction is divided into three stages. The first stage, *registration*, is to iteratively estimate the scale and the 6-dimensional (6D) rigid registration transformation between the input sparse points and the mean shape of the DS-PDM. The second stage, *morphing*, is to optimally and robustly estimate a patient-specific template shape from the DS-PDM using Mahalanobis distance based regularization. The estimated patient-specific template shape is then fed to the third stage, *deformation*, where a newly formularized thin-plate spline kernel-based regularization is used to further reduce the reconstruction error.

The remainder of this paper is organized as follows. Section 2 describes the construction of the DS-PDM from an already-aligned sparse training shape set. Section 3 presents the proposed three-stage reconstruction method using the DS-PDM. Section 4 presents our preliminary study using two plastic bones and the results, followed by the discussions and conclusions in section 5.

2 Construction of Dense Surface Point Distribution Model Using Subdivision

The input data set for this step is the training shape database described in our previous work [13], which consists of 13 segmented proximal femoral surface data. Each individual surface is described by a sparse triangle mesh list containing 4098 vertices. A sequence of correspondence establishing methods was employed to optimally align these training shapes for computing optimal PCA models. It starts with a SPHARM-based parametric surface description [16] and then is optimized using Minimum Description Length (MDL) based principle as proposed by Davies [17].

The basic idea of subdivision is to provide a smooth limit surface which approximates the input data. Starting from a low resolution control mesh, the limit surface is approached by recursively tessellating the mesh. The positions of vertices created by tessellation are computed using a weighted stencil of local vertices. The complexity of the subdivision surface can be increased until it satisfies the user's requirement.

For our purpose, we use a simple subdivision scheme called *Loop scheme*, invented by Charles Loop [18], which is based on a spline basis function, called the three-dimensional quartic box spline. The reasons why we choose *Loop scheme* are that it is defined for triangle mesh, as shown in Fig. 1, and that it guarantees that the limit surface is smooth. Its subdivision principle is very simple. Three new vertices are inserted to divide a triangle in low resolution to four triangles in high resolution.

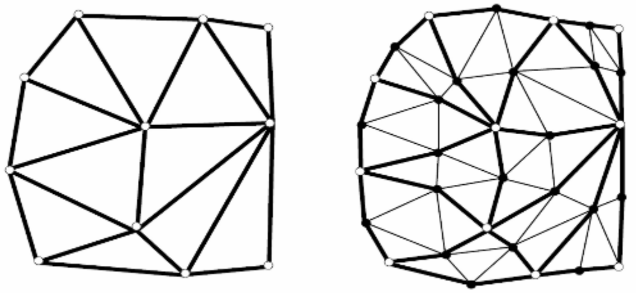


Fig. 1. Subdivision of triangle mesh using *Loop scheme*. Left: original mesh in low resolution; right: subdivided mesh in high resolution. Original vertices are shown as white dots and newly inserted vertices are shown as black dots.

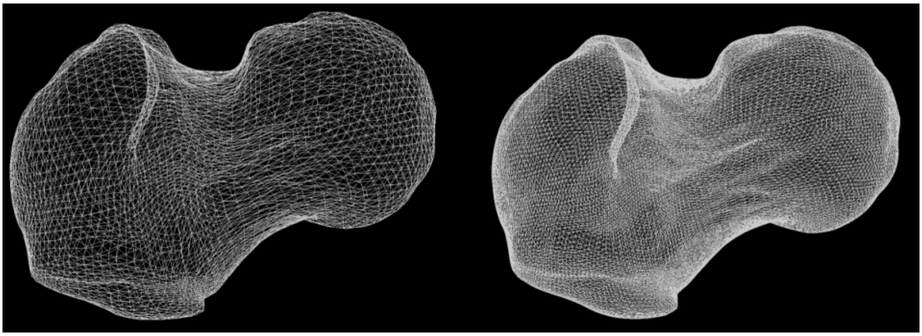


Fig. 2. Subdivision example for one of the surface in the training database. Left: original mesh described with 4098 vertices; right: subdivided mesh described with 16386 vertices. The maximum edge length of all triangles on the subdivided surface is less than 1.5 mm.

As mentioned before, the level of subdivision is depends on the user's requirement. In our case, we require that the maximum edge length of all triangles is less than 1.5 mm. By saying that, a 1-level subdivision is enough for our purpose, which results in totally 16386 vertices per training surface. One of the examples is given by Fig. 2.

The positions of vertices on the control mesh in low resolution are not changed by the Loop subdivision. Furthermore, positions of the inserted vertices in fine resolution are interpolated from the neighboring control vertices in coarse resolution. As the input sparse training surfaces have already been optimized for establishing correspondence, it is reasonable to conclude that the dense surfaces obtained by single-level subdivision are also aligned.

Following that, the DS-PDM is constructed as follows. Let $\mathbf{x}_i = (p_0, p_1, \ldots\ldots, p_{N-1})$, *i = 0, 1, …, m-1* be *m* (here *m* = 13) members of the aligned training population. Each member is described by individual vectors $\mathbf{x}_i$ containing *N* (here *N* = 16386) aligned 3D point coordinates. A statistical shape model is constructed using PCA as follows.

$$
\begin{aligned}
&D = ((m-1)^{-1}) \cdot \textstyle\sum_{i=0}^{m-1} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \\
&P = (\mathbf{p}_0, \mathbf{p}_1, \ldots); \qquad D \cdot \mathbf{p}_i = \sigma_i^2 \cdot \mathbf{p}_i
\end{aligned}
\tag{1}
$$

where $\bar{\mathbf{x}}$ and D represents the mean vector and the covariance matrix respectively. The sorted eigenvalues λ_i and corresponding eigenvectors $\mathbf{p}_i$ of the covariance matrix are the principal directions spanning a shape space with $\bar{\mathbf{x}}$ representing its origin, which itself is a dense surface.

3 The Proposed Three-Stage Reconstruction Technique

Given the positions of a reduced number $n << N$ of intra-operatively digitized points in Euclidean space, $\mathbf{v}' = \{v_i' = (x_i', y_i', z_i'); \quad i = 0, 1, \ldots, n-1\}$, the reconstruction problem is solved in three stages:

1. The *Registration*: this is the only stage solved by iteration. In this stage, the scale and the rigid 6D registration transformation between the mean shape and the input digitized points need to be iteratively determined;
2. *Morphing*: using the estimated scale and pose information from stage 1, a patient template shape for stage 3 needs to be optimally and robustly estimated from DS-PDM by morhing;
3. *Deformation*: the estimated template shape from stage 2 is further deformed to reduce the reconstruction error.

3.1 Registration

This is a well-known problem and several efforts have been made to solve it. One of the most popular methods is the *Iterative Closest Point* (ICP) algorithm developed by Besl and McKay [19], Chen and Medioni [20], and Zhang [21]. The ICP is based on the search of pairs of closest points, and computing a paired-point matching transformation. Then the obtained transformation is applied to one set of points, and the procedure is iterated until convergence. Normally, when trying to register a set of points to a surface described by a triangle mesh, a computation-intensive point-to-surface distance needs to be computed. However, as the mean shape in our case is described by a dense surface, a simple point-to-point distance is enough for our purpose.

It is well-known that ICP algorithm will converge to a local minimum without a proper initialization. In our case, three anatomical landmarks, i.e., greater trochanter, less trochanter, and femoral notch, are used to initialize the registration procedure, which guarantees the convergence of ICP algorithm.

3.2 Morphing

After registration, we can find the corresponding homologous points of input digitized points on the mean shape. Let's denote those homologous points as $\bar{\mathbf{x}}' = \{(\bar{\mathbf{x}}_j)_i;\ \ 0 \le j \le N-1;\ \ i = 0,1,\ldots,n-1\}$, where $(\bar{\mathbf{x}}_j)_i$ denotes that the *j*th point $\bar{\mathbf{x}}_j$ on the dense smooth mean shape $\bar{\mathbf{x}}$ of the DPDM is the closest point to the *i*th input sparse points v_i' The morphing problem is stated as the minimization of the following cost function:

$$E_{\boldsymbol{\alpha}}(\bar{\mathbf{x}}', \mathbf{v}', \mathbf{x}) = E(\bar{\mathbf{x}}', \mathbf{v}', \mathbf{x}) + \rho * E(\mathbf{x});\quad \mathbf{x} = \bar{\mathbf{x}} + \sum_{k=0}^{m-2} \alpha_k \mathbf{p}_k \tag{2}$$

where α_k is the *m-1* shape parameters that describe the to-be-estimated surface $\mathbf{x}$, $E(\bar{\mathbf{x}}', \mathbf{v}', \mathbf{x})$ is the likelihood energy term and $E(\mathbf{x})$ is the prior energy term (or the stabilization term), used to constrain the estimated shape to a realistic result. ρ is a factor that controls the relative weighting between these two terms.

Likelihood Energy Term: The likelihood is expressed by a measure of the least-squares distance between the digitized points to the predicted shape:

$$E(\bar{\mathbf{x}}', \mathbf{v}', \mathbf{x}) = (n^{-1}) \sum_{i=0}^{n-1} \| v_i' - ((\bar{\mathbf{x}}_j)_i + \sum_{k=0}^{m-2} \alpha_k \cdot \mathbf{p}_k(j)) \|^2; \tag{3}$$

where $\mathbf{p}_k(j)$ is the *j*th tuple of the *k*th shape basis vector.

Prior Energy Term: Due to the PCA construction, the random variable α_k are independent and follow a normal law of a null mean and variance λ_k [5]. To penalize the deviation of the predicted shape from the mean shape, Mahalanobis distance is used as the energy term of this prior model:

$$E(\mathbf{v}) = (1/2) \cdot \sum_{k=0}^{m-2} (\alpha_k^2 / \lambda_k) \tag{4}$$

To determine the shape parameters α_k, the cost function is differentiated with respect to the shape parameters and equated to zero resulting in a linear system of m unknowns, which is solved with standard linear equations system solvers such as LU decompositions.

3.3 Deformation

Similar to the second stage, we also need to find the corresponding homologous points of the input sparse points $\mathbf{v}'$ on the template surface $\mathbf{x}$. Let's denote these homologous points as $\mathbf{v} = \{v_i = (\mathbf{x}_l)_i = (x_i, y_i, z_i);\ \ i = 0,1,...,n-1\}$. The deformation is described as a regression problem of finding a transform $\mathbf{t} = (f, g, h) : \mathbb{R}^3 \to \mathbb{R}^3$ that minimizes following cost function:

$$E(\mathbf{t}) = (n^{-1}) \sum_{i=0}^{n-1} \| v_i' - \mathbf{t}(v_i) \|^2 + (\tau \cdot \mathbf{ln}(m) / \mathbf{ln}(n)) \cdot L[\mathbf{t}] \tag{5}$$

where $\phi[\mathbf{t}] \geq 0$ is a regularization functional, $\tau \geq 0$ is a regularization parameter, m is the number of samples in the training population, n is the number of digitized points, $\mathbf{v}'$ and $\mathbf{v}$ are the n input digitized points and their corresponding homologous points on the morphed shape, respectively. $\mathbf{t}(\mathbf{v}) = \{\mathbf{t}(v_i);\ \ i = 0,1,...,n-1\}$ is the results of applying deformation transform on those homologous points.

From regularization theory, $\phi[\mathbf{t}]$ can be defined as a norm in a reproducing kernel Hilbert space (RKHS) which can be uniquely induced by a positive definite (or conditionally positive definite) kernel function $U(v_i, v_j)$. For our purpose, we propose to use the thin-plate spline (TPS) kernel $U(v_i, v_j) = \| v_i - v_j \|$ in 3D. Now $\phi[\mathbf{t}]$, the bending energy, has the form:

$$\begin{aligned} &\phi[\mathbf{t}] = \iiint_{\mathbb{R}^3} (I(f) + I(g) + I(h)) dxdydz \\ &I(\cdot) = (\frac{\partial^2}{\partial x^2})^2 + 2(\frac{\partial^2}{\partial x \partial y})^2 + (\frac{\partial^2}{\partial y^2})^2 + 2(\frac{\partial^2}{\partial y \partial z})^2 + (\frac{\partial^2}{\partial z^2})^2 + 2(\frac{\partial^2}{\partial z \partial x})^2 \end{aligned} \tag{6}$$

The thin-plate kernel is conditionally positive definite and the affine subspace form the null space of the resulting transform $\mathbf{t} = (f, g, h)$, which must be of the form [22]:

$$\begin{cases} f(v) = a_1 + a_2 x + a_3 y + a_4 z + \sum_{i=0}^{n-1} \gamma_i U(v, v_i) \\ g(v) = b_1 + b_2 x + b_3 y + b_4 z + \sum_{i=0}^{n-1} \theta_i U(v, v_i) \\ h(v) = c_1 + c_2 x + c_3 y + c_4 z + \sum_{i=0}^{n-1} \omega_i U(v, v_i) \end{cases} \tag{7}$$

where $\mathbf{a} = (a_1, a_2, a_3, a_4)^T$, $\mathbf{b} = (b_1, b_2, b_3, b_4)^T$, $\mathbf{c} = (c_1, c_2, c_3, c_4)^T$ represent the affine coefficients and $\boldsymbol{\gamma} = (\gamma_0, ..., \gamma_{n-1})^T$, $\boldsymbol{\theta} = (\theta_0, ..., \theta_{n-1})^T$, $\boldsymbol{\omega} = (\omega_0, ..., \omega_{n-1})^T$ are the kernel interpolation coefficients, And the measure of the smoothness of the nonlinear mapping is given by:

$$L[\mathbf{t}] = \boldsymbol{\gamma}^T \mathbf{K} \boldsymbol{\gamma} + \boldsymbol{\theta}^T \mathbf{K} \boldsymbol{\theta} + \boldsymbol{\omega}^T \mathbf{K} \boldsymbol{\omega} \tag{8}$$

where $k_{ij} = \Phi(v_i, v_j)$; $i, j = 0,1,..., n-1$ are the elements of matrix $\mathbf{K}$.

To determine the affine transformation coefficients $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, and the kernel interpolation coefficients $\boldsymbol{\gamma}$, $\boldsymbol{\theta}$, $\boldsymbol{\omega}$, the cost function is differentiated with respect to all these transform parameters and equated to zero resulting in following linear system:

$$\begin{pmatrix} \mathbf{x}' & \mathbf{y}' & \mathbf{z}' \\ \mathbf{o} & \mathbf{o} & \mathbf{o} \end{pmatrix} = \begin{pmatrix} (\mathbf{K} + (\tau \cdot \mathbf{ln}(m)/\mathbf{ln}(n)) \cdot \mathbf{I}) & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O} \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma} & \boldsymbol{\theta} & \boldsymbol{\omega} \\ \mathbf{a} & \mathbf{b} & \mathbf{c} \end{pmatrix} \tag{9}$$

where $\mathbf{O}$ is a *4* x *4* matrix of zeros, $\mathbf{o}$ is a *4* x *1* vectors of zeros. **P = (1, x, y, z)**, where $\mathbf{1} = (1, ..., 1)^T$. Note that $\mathbf{x} = (x_0, ..., x_{n-1})^T$, $\mathbf{y} = (y_0, ..., y_{n-1})^T$, $\mathbf{z} = (z_0, ..., z_{n-1})^T$; $\mathbf{x}' = (x_0', ..., x_{n-1}')^T$, $\mathbf{y}' = (y_0', ..., y_{n-1}')^T$, $\mathbf{z}' = (z_0', ..., z_{n-1}')^T$ represents coordinates of the input points and their corresponding points on the template surface, respectively.

4 Preliminary Study and Results

To verify the effectiveness of the proposed three-stage method, a preliminary study on reconstruction of two plastic femoral heads was performed. We call one of the plastic bones as "No.1" and the other as"No. 2", (see Fig. 3). For both bones, we have designed four different studies using 10, 20, 40, 80 points respectively. To eliminate other sources of errors such as digitization error or matching error, those points are directly extracted from the surfaces of the bones, which are accurately segmented from the corresponding CT volume scans using commercially available software Amira™ (Mercury Computer Systems Inc., Germany). And in each study, results obtained in two sequential experiments are recorded: experiment 1, morphing after registration; and experiment 2, deformation after morphing. The reconstructed surface in each experiment is directly compared to the segmented surfaces from CT volume data, which we take them as the ground truth. Symmetric Hausdorff Distance is employed to measure the distance between discrete 3D surfaces [23].

The results of the preliminary study are listed in Table 1. It was found that morphing after registration could already give a reasonable accurate result for the No.1 bone even with a small number of points, when the DS-PDM was used. But the dramatic increase of the number of points did not result in a similar increase of accuracy in this stage. On the contrast, including the third stage into the reconstruction resulted in a steady increase of accuracy for both cases. In all experiments, we have chosen τ = 0.5. The parameter ρ was chosen as following principle: when more points were received, ρ was changed to smaller to relax the Mahalanobis distance term.

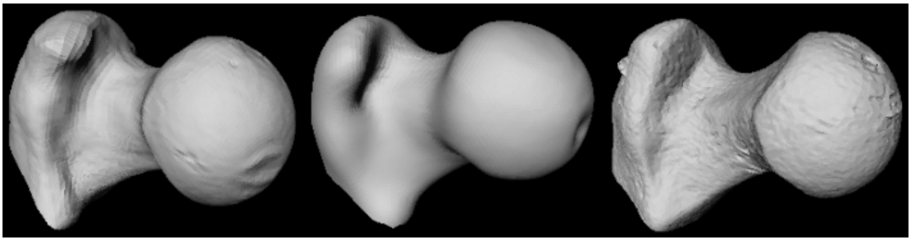
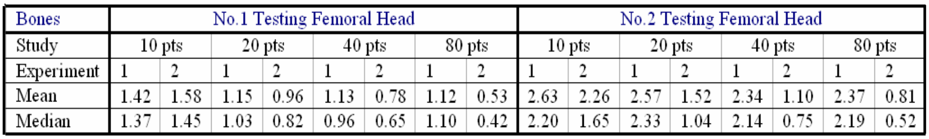


Fig. 3. Surface rendering of testing femoral heads and mean shape of the DS-PDM. Left: No.1 testing femoral head; Middle: mean shape of the DS-PDM; Right: No.2 testing femoral head.

Table 1. Results of using different number of points for surface reconstruction (unit: mm)



Bones	No.1 Testing Femoral Head								No.2 Testing Femoral Head							
Study	10 pts		20 pts		40 pts		80 pts		10 pts		20 pts		40 pts		80 pts	
Experiment	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Mean	1.42	1.58	1.15	0.96	1.13	0.78	1.12	0.53	2.63	2.26	2.57	1.52	2.34	1.10	2.37	0.81
Median	1.37	1.45	1.03	0.82	0.96	0.65	1.10	0.42	2.20	1.65	2.33	1.04	2.14	0.75	2.19	0.52

5 Discussions and Conclusions

It is very interesting to find that morphing after registration can already give a reasonably accurate result but doesn't improve too much as more and more points is added. This may be explained by the correlation between vector components which is implicitly stored in the statistical deformable model. On the contrast, starting from the morphed surface, the deformation step improves accuracy greatly as more and more points are added. This can be well explained by the RKHS theory, as more points mean higher the dimensionality of the RKHS, derived from the conditionally positive definite matrix **K** [4].

Using DS-PDM facilitates the whole surface reconstruction process. The increase of the number of vertices doesn't necessarily mean a dramatic increase of computation time, as more efficient data structure such as k-D tree could be employed [23]. The smoothly and densely described *a priori* information causes the whole reconstruction procedure more robust and more accurate.

Our formularization of the regression technique as given by equation (5) is optimal. It allows the proposed method to adapt gradually either to use more information derived from the statistical shape model when larger training data are available or to use more information derived from the digitized points when more points are input. On the extreme cases, it equals to an exact TPS interpolation when only one sample of training data is available, or equals to an affine registration when infinite training samples are available.

References

1. Lavalle S., Merloz P., et al. Echomorphing introducing an intra-operative imaging modality to reconstruct 3d bone surfaces for minimally invasive surgery. CAOS 2004, pp. 38 – 39
2. Hofstetter R., Slomczykowski M, et al. Fluoroscopy as an imaging means for computer-assisted surgical navigation. Comp Aid Surg Vol. 4, pp. 65 – 76, 2004

3. Livyatan H.m Yaniv Z, Joskowicz J. Gradient-based 2-D/3-D rigid registration of fluoroscopic X-ray to CT. IEEE T Med Imaging. Vol 22, pp. 1395 – 1406, 2004
4. Evgeniou T., Pontil M., Poggio T., Regularization networks and support vector machines. Adv Comput Math, vol. 13, pp. 1-50, 2000
5. Cootes T. F., Taylor C. J., Cooper D. H., Graham J. Active shape models – their training and application. Comput Vis Image Und, vol. 61, no.1, pp. 38-59, 1995
6. Golland P., Grimson W. E. L., Shenton M. E., Kikinis R. Small sample size learning for shape analysis of anatomical structures. MICCAI 2000, pp. 72-82, 2000
7. Blanz V. and Vetter T. Reconstructing the complete 3D shape of faces from partial information. it+ti Oldenburg Verlag, pp. 295-302, 2002
8. Blanz V. and Vetter T. A morphable model for the synthesis of 3d faces. SIGGRAPH 1999, pp. 187 – 194
9. Lapeer R. J. A., Prager R. W. 3D shape recovery of a newborn skull using thin-plate splines. Comput Med Imag Grap, vol. 24, pp. 193–204, 2000
10. Fleute M. and Lavallee S. Building a complete surface model from sparse data using statistical shape models: application to computer assisted knee surgery system. MICCAI 1998, pp. 879-887, 1998
11. Stindel E., Briard J. L. et al. Bone morphing: 3D Morphological data for total knee arthroplasty. Comp Aid Surg. Vol. 7, pp. 156 – 168, 2002
12. Chan C. S., Edwards P. J., Hawkes D. J. Integration of ultrasound-based registration with statistical shape models for computer-assisted orthopedic surgery. SPIE, Medical Imaging, pp. 414-424, 2003
13. Rajamani T. K., Nolte L.-P., Styner M. Bone morphing with statistical models fro enhanced visualization. SPIE Medical Imaging, pp. 122 – 130, 2004
14. Rajamani T. K., Joshi S., Styner M. Bone model morphing for enhanced surgical visualization. IEEE International Symposium on Biomedical Imaging, pp. 1255 – 1258, 2004
15. Rajamani T. K., et al. A novel and stable approach to anatomical structure morphing for enhanced intraoperative 3D visualization. SPIE Medical Imaging, pp. 718 – 725, 2005.
16. Brechbuehler C., Gerig G., Kuebler O. Parameterization of closed surfaces for 3D shape description. Comput Vis Image Und
17. Davies R. H., Twining C. H., et al. 3D statistical shape models using direct optimization of description length. ECCV 2002, pp. 3 – 20, 2002.
18. Loop C. T. Smooth subdivision surfaces based on triangles. M.S. Thesis, Department of Mathematics, University of Utah, August 1987
19. Besl P. J., McKay N. D. A method for registration of 3-D shapes. IEEE T Pattern Anal, vol. 14, pp. 239 – 256, 1992.
20. Chen Y. and Medioni G. Object modeling by registration of multiple range images. Image Vision Comput. Vol 10, pp. 145 – 155, 1992
21. Zhang Z. Iterative point matching for registration of free-form curves and surfaces. Int J Comput Vision. Vol 13, pp. 119 – 152, 1994
22. Bookstein F. Principal warps: thin-plate splines and the decomposition of deformations. IEEE T Pattern Anal. Vol 11., pp. 567 – 585, 1989
23. Aspert N., Santa-Cruz D., Ebrahimi T. MESH: Measuring errors between surfaces using the Hausdorff Distance. IEEE International Conference on Multimedia and Expo (ICME) 2002. pp. 705 – 708, 2002

Fisheye Lenses Calibration Using Straight-Line Spherical Perspective Projection Constraint

Xianghua Ying[1,⋆], Zhanyi Hu[2], and Hongbin Zha[1]

[1] National Laboratory on Machine Perception, Peking University, Beijing, P.R. China
{xhying, zha}@cis.pku.edu.cn
http://www.cis.pku.edu.cn/vision/Visual&Robot/people/ying/
[2] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences
huzy@nlpr.ia.ac.cn

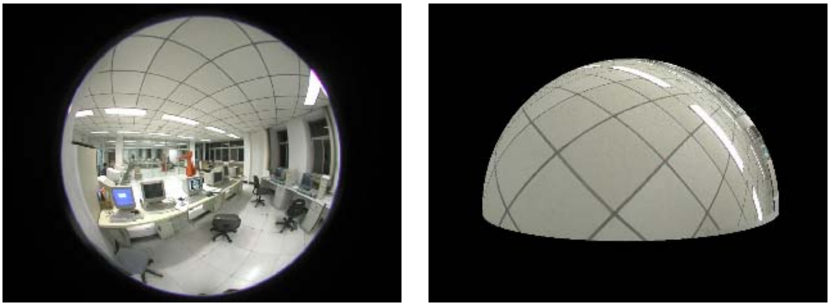
Abstract. Fisheye lenses are often used to enlarge the field of view (FOV) of a conventional camera. But the images taken with fisheye lenses have severe distortions. This paper proposes a novel calibration method for fisheye lenses using images of space lines in a single fisheye image. Since some fisheye cameras' FOV are around 180 degrees, the spherical perspective projection model is employed. It is well known that under spherical perspective projection, straight lines in space have to be projected into great circles in the spherical perspective image. That is called straight-line spherical perspective projection constraint (SLSPPC). In this paper, we use SLSPPC to determine the mapping between a fisheye image and its corresponding spherical perspective image. Once the mapping is obtained, the fisheye lenses is calibrated. The parameters to be calibrated include principal point, aspect ratio, skew factor, and distortion parameters. Experimental results for synthetic data and real images are presented to demonstrate the performances of our calibration algorithm.

1 Introduction

In many computer vision applications, including robot navigation, 3D reconstruction, image-based rendering, and single view metrology, a camera with a quite large field of view (FOV) is preferable. A conventional camera has a very limited FOV. Therefore, cameras with wide-angle or fisheye lenses are often employed. Images taken with these imaging devices often have significant distortions. If we want to use some perspective information from these distorted images, they have to be transformed into perspective images. A fisheye camera's FOV is around 180 degrees, but a wide-angle camera's FOV is usually around 100 degrees. The existing calibration methods [2, 4, 5, 9] for wide-angle camera using images of space lines cannot be directly used for fisheye cameras. Therefore, this paper aims at calibrating fisheye cameras using images of space lines. An image from a fisheye camera with FOV 183 degrees (Nikon COOLPIX 990 with FC-E8 fisheye lenses) is shown in Fig. 1a.

[⋆] This work was partially carried out while the author was at the Chinese Academy of Sciences.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 61–70, 2006.



(a) (b)

Fig. 1. (a) A fisheye image. (b) The corresponding spherical perspective projection image. The calibration procedure is to find the mapping between those two.

In literature, there are two standard types of perspective images used in computer vision: planar and spherical surfaces (i.e., a planar or a spherical surface can be used as the retina of a perspective camera). Due to lens distortions, space lines are projected into image curves in the actual image. Once the mapping between a distorted image and its corresponding perspective image is obtained, the calibration problem is solved. The mapping can be obtained by finding the relation between the image curves of space lines and its corresponding perspective images. It is well known that under planar perspective projection, images of straight lines in space have to be mapped into straight lines in the planar perspective image. That is called the straight-line planar perspective projection constraint (SLPPPC). The existing calibration methods [2, 4, 5, 9] for wide-angle cameras using the distorted images of lines are all based on SLPPPC. However, for fisheye cameras with FOV around 180 degrees, we use the spherical perspective projection model because it is a convenient way to represent FOV around 180 degrees. We also know that under spherical perspective projection, images of straight lines in space have to be projected into great circles in the spherical perspective image. Therefore, there exists another constraint we called the straight-line spherical perspective projection constraint (SLSPPC). In this paper we elaborate on how to determine the mapping between a fisheye image and its corresponding spherical perspective image using SLSPPC (see Fig. 1).

2 Fisheye Imaging Model

Fisheye imaging model describes a mapping from 3D space points to 2D fisheye image points (see Fig. 2). We introduce the spherical perspective projection into the fisheye imaging model and divide the imaging model into four concatenated steps as follows:

Step 1: Transform the 3D world coordinates of a space point into the 3D camera coordinates.

Considering a generic 3D point, visible by a fisheye camera, with Cartesian coordinates $\mathbf{P}_W = (X, Y, Z)^T$ in the world coordinate system, if $\mathbf{P}_C =$

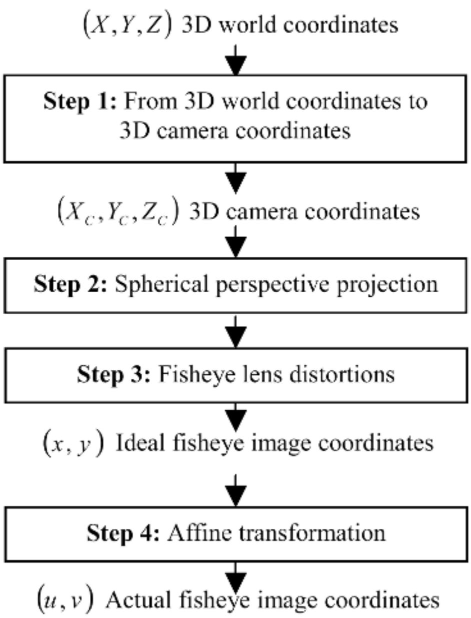

(X,Y,Z) 3D world coordinates
Step 1: From 3D world coordinates to 3D camera coordinates
(X_C,Y_C,Z_C) 3D camera coordinates
Step 2: Spherical perspective projection
Step 3: Fisheye lens distortions
(x, y) Ideal fisheye image coordinates
Step 4: Affine transformation
(u, v) Actual fisheye image coordinates

Fig. 2. Fisheye imaging model

$(X_C, Y_C, Z_C)^T$ are the coordinates in the camera coordinate system, the transformation between $\mathbf{P}_W$ and $\mathbf{P}_C$ is:

$$\mathbf{P}_C = \mathbf{R}\mathbf{P}_W + \mathbf{t}, \tag{1}$$

where the matrix $\mathbf{R}$ and vector $\mathbf{t}$ describe the orientation and position of the fisheye camera with respect to the world coordinate system. The parameters in $\mathbf{R}$ and $\mathbf{t}$ are called the extrinsic parameters.

Step 2: The space point is perspectively projected onto a unit sphere centered at the projection center. This procedure can be represented by a transformation from the 3D camera coordinates to the 2D spherical coordinates.

The unit sphere is called the viewing sphere. If $\mathbf{p}$ is the spherical projection of the space point, we have:

$$\mathbf{p} = \frac{\mathbf{P}_C}{\|\mathbf{P}_C\|} = (\sin\Phi\cos\Theta, \sin\Phi\sin\Theta, \cos\Phi)^T, \tag{2}$$

where $\mathbf{p} = (\sin\Phi\cos\Theta, \sin\Phi\sin\Theta, \cos\Phi)^T$ is the unit directional vector, and (Φ, Θ) is the 2D spherical coordinates of the spherical point (see Fig. 3). Obviously, (Φ, Θ) can be determined from $\mathbf{p}$, and vice versa.

Step 3: The spherical projection point $\mathbf{p}$ is mapped to $\mathbf{m}$ on the image plane due to fisheye lens distortions, which can be represented as:

$$\mathbf{m} = D(\mathbf{p}), \tag{3}$$

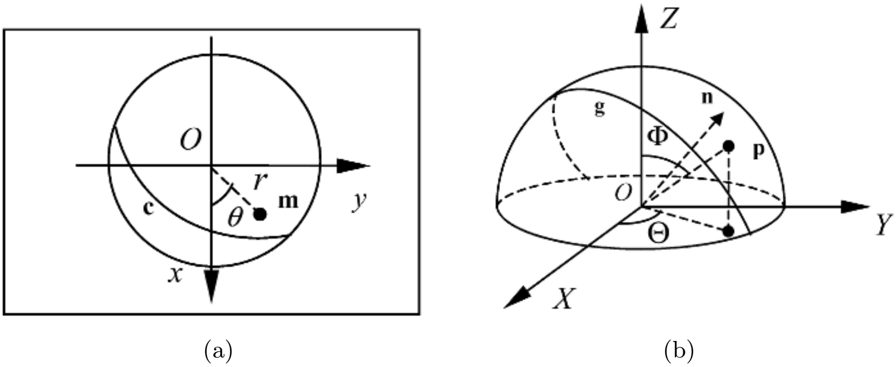


Fig. 3. (a) An ideal fisheye image. (b) The corresponding spherical perspective image. The spherical point **p** is mapped to **m** in the ideal fisheye image using the fisheye distortion model D. The great circle **g** which is the spherical projection of a straight line in space is mapped to a image curve **c** in the ideal fisheye image also using the fisheye distortion model D.

where $\mathbf{m} = (x, y)$, and D is the so-called fisheye distortion model. The image obtained here is called the ideal fisheye image. The parameters in D are called the distortion parameters. The fisheye distortion model will be discussed in details in the next section. Note that in Step 3, we obtain a planar image with the pixel coordinates, where the origin of the image coordinate system is located at the principal point, and the image coordinate system has equal scales in the directions of two coordinate axes.

Step 4: The image point **m** is transformed into **m**′ using an affine transformation:

$$\mathbf{m}' = \mathbf{K}_A(\mathbf{m}), \tag{4}$$

where $\mathbf{m}' = (u, v)$. The image obtained here is called the actual fisheye image. The meaning of formula (4) is:

$$\tilde{\mathbf{m}}' = \mathbf{K}_A \tilde{\mathbf{m}}, \tag{5}$$

where $\tilde{\mathbf{m}} = (x, y, 1)^T$ and $\tilde{\mathbf{m}}' = (u, v, 1)^T$ are the homogeneous coordinates corresponding to **m** and **m**′ respectively, and

$$\mathbf{K}_A = \begin{bmatrix} r & s & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

3 Fisheye Distortion Model

Fisheye distortion model D describes the mapping from a spherical perspective image to its corresponding ideal fisheye image (see Fig. 2 and Fig. 3). If **p** is the spherical perspective projection of a space point and (Φ, Θ) are the 2D spherical coordinates of $\mathbf{p}$, due to fisheye lens distortions, $\mathbf{p}$ is mapped to $\mathbf{m}$ in the ideal fisheye image. If (x, y) is the Cartesian coordinates and (r, θ) is the polar coordinates of where the origins of the two coordinate systems are both located at the principal point, the relation between (x, y) and (r, θ) is:

$$r = \sqrt{x^2 + y^2}, \tan\theta = \frac{y}{x}. \tag{7}$$

In our experiments, we use fifth degree polynomials to represent fisheye radial and tangential distortion models:

$$r = D_R(\Phi) = \sum_{i=1}^{5} d_i \Phi^i, \theta = D_T(\Theta) = \sum_{i=1}^{5} b_i \Theta^i, \tag{8}$$

where d_i are radial, and b_i are tangential distortion parameters. In fact, any other proper parametric distortion models for fisheye lenses can be employed, such as those proposed in [1, 3, 7, 8, 10, 11].

Since the FOV of fisheye lenses is known, we have:

$$\gamma = D_R(\frac{\alpha}{2}), \tag{9}$$

where γ is the radius of the ideal fisheye image, α is the fisheye lenses' FOV. After some manipulation, we have:

$$d_5 = \frac{32\gamma - 16\alpha d_1 - 8\alpha^2 d_2 - 4\alpha^3 d_3 - 2\alpha^4 d_4}{\alpha^5}. \tag{10}$$

So there are only four independent parameters for radial distortion. The longitude angle and the polar direction are both periodic. From $\Theta = 0$ and (8), we have $\theta = 0$. Therefore, if $\Theta = 2\pi$, then $\theta = 2\pi$. Thus we have:

$$b_5 = \frac{1 - b_1 - 2\pi b_2 - 4\pi^2 b_3 - 8\pi^3 b_4}{16\pi^4}. \tag{11}$$

So there are only four independent parameters for tangential distortion.

4 Fisheye Camera Calibration

From the discussions above, we know that there are totally 12 parameters for a fisheye lenses required to be calibrated: 4 affine transformation parameters, 4 radial and 4 tangential distortion parameters. These parameters are called the extended intrinsic parameters in this paper.

Given a fisheye image containing several image curves of space lines, we select a small set of points along these image curves. These sample points are mapped to spherical points on the viewing sphere using the concatenation of $\mathbf{K}_A^{-1}$ and D^{-1} , and the great circle fitting method is employed. The objective function is the sum of the squared distances of these spherical points from their corresponding best-fit great circles. In this section, we firstly introduce the algorithm for great circle fitting. Secondly, the objective function with the extended intrinsic parameters is constructed, and finally, how to find the initial values for these parameters is discussed.

4.1 Great Circle Fitting

A great circle is the intersection of a sphere and a plane passing through the spherical center. It can be determined by two parameters (α, β) which are the directional angles of the normal vector for the plane containing the great circle in the 3D Cartesian coordinate system whose origin is located at the spherical center (see Fig. 3b). For a spherical point $\mathbf{p}$ and a great circle $\mathbf{g} = (\alpha, \beta)$ where the unit normal vector for the plane containing the great circle is $\mathbf{n} = (\sin\alpha\cos\beta, \sin\alpha\sin\beta, \cos\alpha)^T$, the distance from $\mathbf{p}$ to the plane containing the great circle is $d = |\mathbf{p}^T\mathbf{n}|$. As noted in [6], the great circle fitting problem may be replaced by the problem of finding a plane so as to minimize the sum of squares of distances between the given points and the plane. Given N spherical points $\mathbf{p}_i$, the objective function is constructed as the sum of the squared distances of $\mathbf{p}_i$ from the plane containing the best-fit great circle:

$$F(\mathbf{n}) = \sum_{i=1}^{N} (\mathbf{p}_i^T\mathbf{n})^2, \tag{12}$$

where $\mathbf{n}$ is the normal vector for the plane. This can be converted into an eigenvalue problem. A vector equation is introduced as:

$$\mathbf{An} = 0, \tag{13}$$

where $\mathbf{A} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N)^T$. The objective function becomes:

$$F(\mathbf{n}) = (\mathbf{An})^T\mathbf{An} = \mathbf{n}^T\mathbf{Bn}. \tag{14}$$

The solution for $\mathbf{n}$ is the eigenvector of $\mathbf{B}$ corresponding to the smallest eigenvalue. $\mathbf{g} = (\alpha, \beta)$ can be easily computed from the obtained $\mathbf{n}$.

4.2 Objective Function Formulation

We use L to represent the number of the sample image curves of space lines in the actual fisheye image, and use $N_j (j = 1, \ldots, L)$ to represent the number of the sample points on the j^{th} image curve. $\mathbf{m}'_{i,j} (j = 1, \ldots, L)$ represents the image coordinates of the sample point on the j^{th} image curve. The objective function can be constructed as:

$$\xi = \sum_{j=1}^{L} F(\mathbf{n}_j) = \sum_{j=1}^{L} \left[\sum_{i=1}^{N_j} (\mathbf{p}_{i,j}^T\mathbf{n}_j)^2 \right], \tag{15}$$

where $\mathbf{n}_j = (\sin\alpha_j\cos\beta_j, \sin\alpha_j\sin\beta_j, \cos\alpha_j)^T$ is the normal vector for the plane containing the best-fit great circle $\mathbf{g}_j = (\alpha_j, \beta_j)$, and

$$\mathbf{p}_{i,j} = D^{-1}(\mathbf{K}_A^{-1}(\mathbf{m}'_{i,j})), \tag{16}$$

where $\mathbf{p}_{i,j}$ represents the spherical point obtained from the sample image point $\mathbf{m}'_{i,j}$ after using the concatenation of $\mathbf{K}_A^{-1}$ and D^{-1} . The objective function ξ

describes the sum of the squared distances of $\mathbf{p}_{i,j}$ from its corresponding best-fit great circle $\mathbf{g}_j$. The Levenberg-Marquardt optimization technique is used to perform this minimization. The parameters for the great circles $\mathbf{g}_j = (\alpha_j, \beta_j)(j = 1, \ldots, L)$ are optimized together with the extended intrinsic parameters. As we know the initial values for the optimized parameters are required in the nonlinear minimization, therefore, the initial estimations of these parameters are discussed in the next section.

4.3 Initial Estimations

Affine Transformation Parameters. A significant characteristic of an actual fisheye image is that its boundary is usually an ellipse (see Fig. 1a). In fact, the bounding ellipse is the projection of the boundary between the optical components (glass lenses) and their metal supporting part. Light rays are occluded by the supporting part when the light rays are out of the fisheye camera FOV. The shape of the physical boundary is a circle. The optical axis of the fisheye camera is perpendicular to the plane containing the circle, and it also goes through the center of the circle. To identify the bounding ellipse of the fisheye image, we use a predefined threshold to find the boundary, and fit an ellipse to the resulting boundary. If the equation of the bounding ellipse is:

$$a'u^2 + 2b'uv + c'v^2 + 2d'u + 2e'v + f = 0, \tag{17}$$

we may obtain the initial values for affine transformation parameters as:

$$\begin{array}{ll} \overline{r} = \sqrt{-\frac{b'^2}{a'^2} + \frac{c'}{a'}} & \overline{s} = -\frac{b'}{a'} \\ \overline{u}_0 = \frac{b'e' - c'd'}{a'c' - b'^2} & \overline{v}_0 = \frac{b'd' - a'e'}{a'c' - b'^2} \end{array} . \tag{18}$$

Due to lack of space, the derivation is omitted here.

Distortion Correction Parameters. Since the equidistance model is a very good approximation to the real radial distortion of a fisheye camera [10], the initial values of the radial distortion correction parameters are set as: $\overline{c}_1 = \frac{\alpha}{2\gamma}$, and $\overline{c}_2 = \overline{c}_3 = \overline{c}_4 = 0.0$, where γ is the radius of the ideal fisheye image and α is the fisheye camera FOV. For the tangential distortion, the reasonable initial values are $\overline{a}_1 = 1.0, \overline{a}_2 = \overline{a}_3 = \overline{a}_4 = 0.0$ (i.e., $\Theta = \theta$).

Parameters of Best-Fit Great Circles. When the initial values for the extended intrinsic parameters have been obtained, we have:

$$\overline{\mathbf{p}}_{i,j} = \overline{D}^{-1}(\overline{\mathbf{K}}_A^{-1}(\mathbf{m}'_{i,j})), \tag{19}$$

where $\overline{\mathbf{K}}_A^{-1}$ and $\overline{D}^{-1}$ are $\mathbf{K}_A^{-1}$ and D^{-1} with the initial parameters respectively. $\overline{\mathbf{p}}_{i,j}$ represents the spherical point obtained from the sample image point $\mathbf{m}'_{i,j}$ after using the concatenation of $\overline{\mathbf{K}}_A^{-1}$ and $\overline{D}^{-1}$. From these spherical points $\overline{\mathbf{p}}_{i,j}$, the great circle fitting method described in Sect. 4.1 is used to fit great circles $\overline{\mathbf{g}}_j = (\alpha_j, \beta_j)(j = 1, \ldots, L)$ respectively. Therefore, the initial values for the parameters of the best-fit great circles are obtained.

5 Experiments

5.1 Simulations

We have performed a number of experiments with simulated data in order to assess the performances of our calibration algorithm. The extended intrinsic parameters $\{r, s, u_0, v_0, c_1, c_2, c_3, c_4, a_1, a_2, a_3, a_4\}$ for the simulated fisheye camera are generated randomly distributed within their corresponding valid ranges. The simulated fisheye lenses FOV is 180 degrees. The resolution of the image is 1024×1024. The generation procedure is constructed as follows: Firstly, the great circles are generated which representing the spherical projection of straight lines in space. Secondly, these great circles are transformed into image curves using D and $\mathbf{K}_A$. Thirdly, on each image curve about 50 points are chosen. Gaussian noise with zero-mean and σ standard deviation is added to these image points. The noise level σ is varied from 0.2 to 2.0 pixels. Finally, the ellipse boundary is also generated in the simulated fisheye image (see Fig. 4a).

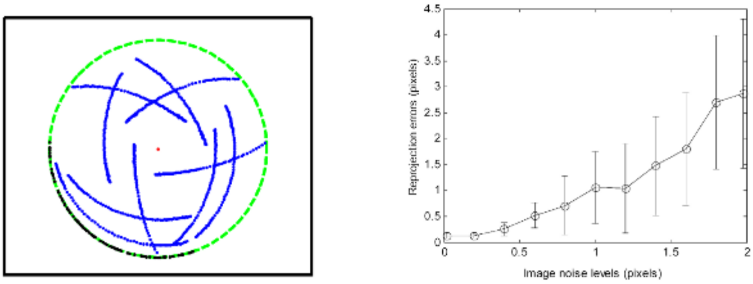
In order to compare the recovered parameters with the ground truth, similar to [9], we use the reprojection error to evaluate the calibration accuracy:

$$\varepsilon_{rep} = \frac{1}{\sum_{j=1}^{L} N_j} \sum_{j=1}^{L} \left[\sum_{i=1}^{N_j} \|\mathbf{m}'_{i,j} - \mathbf{K}_A D(\widehat{D}^{-1}\widehat{\mathbf{K}}_A^{-1}(\mathbf{m}'_{i,j}))\| \right] , \qquad (20)$$

where $\mathbf{m}'_{i,j}$ are the coordinates of the sample points in the simulated fisheye image. $\mathbf{K}_A$ and D are with the ground truth. $\widehat{D}^{-1}$ and $\widehat{\mathbf{K}}_A^{-1}$ are with the recovered values. For each noise level, we perform 1000 independent trials, and the reprojection errors are computed over each run. The means and standard deviations of reprojection errors with respect to different noise levels are shown in Fig. 4b.

5.2 Real Images

The fisheye lenses used here is Nikon FC-E8 with FOV 183 degrees, mounted on a Nikon COOLPIX 990 digital camera. A fisheye image taken with this fisheye



(a) (b)

Fig. 4. Simulation results for fisheye calibration. (a) A simulated fisheye image containing image curves of straight lines in space. (b) The means and the standard deviations of the reprojection errors with respect to different noise levels.

Table 1. The mean and maximum of the reprojection errors for the three planar homographies. The errors shown here are divided by the side length of a square grid.

	Mean error	Max. error
$\{\mathbf{x} \leftrightarrow \mathbf{q}_1\}$	76.44%	246.92%
$\{\mathbf{x} \leftrightarrow \mathbf{q}_2\}$	13.21%	40.96%
$\{\mathbf{x} \leftrightarrow \mathbf{q}_3\}$	1.64%	3.08%

camera for calibration is shown in Fig. 1a. The resolution of the fisheye image is 2048×1536. From this fisheye image, about 10 image curves of the straight lines in space and total about 500 sample points are chosen. The extended intrinsic parameters of the fisheye camera are recovered using our calibration method. Then, we apply these recovered parameters to undistort the fisheye image, and the spherical perspective image is obtained as shown in Fig. 1b.

Here, we use planar homography constraint to evaluate calibration accuracy. We select some fisheye images of grid points on a ceiling in Fig. 1a. There are totally three sets of point pairs for evaluating the homography constraint: $\{\mathbf{x} \leftrightarrow \mathbf{q}_1\}$, $\{\mathbf{x} \leftrightarrow \mathbf{q}_2\}$ and $\{\mathbf{x} \leftrightarrow \mathbf{q}_3\}$, where $\mathbf{q}_1$ represents the 2D homogeneous coordinates of the fisheye image point, $\mathbf{q}_2$ represents the unit directional vector of the spherical point obtained using the distortion correction procedure with the initial values of the extended intrinsic parameters, and $\mathbf{q}_3$ similar to $\mathbf{q}_2$ but with the recovered values. The reprojection error to evaluate homography constraint is:

$$\varepsilon_i = \|\mathbf{x} - \mathbf{H}_i^{-1}\mathbf{q}_i\|, i = 1, 2, 3, \tag{21}$$

where $\mathbf{H}_i (i = 1, 2, 3)$ is the obtained planar homography. The mean and maximum of the reprojection errors are shown in Table 1. From Table 1, we can see that the improvement of the planar homography constraint is very significant due to the fisheye lenses distortion correction.

6 Conclusions

In this paper, we propose a novel calibration method for fisheye lenses using the images of space lines. The SLSPPC is employed for calibrating fisheye lenses with FOV around 180 degrees, whereas the existing methods based on SLPPPC cannot be used in this case. The extended intrinsic parameters of fisheye cameras can be calibrated without needing to seek the extrinsic parameters. Thus, the number of parameters to be calibrated is drastically reduced, making the calibration procedure simple and practical. Our method can use any other suitable parametric distortion models for fisheye lenses though we only use the polynomial models here.

Acknowledgements

This work was supported in part by the NSFC Grant (No. 60333010), and NKBRPC (No. 2004CB318000).

References

1. A. Basu and S. Licardie: Alternative models for fish-eye lenses, Pattern Recognition Letters, 16(4), 1995, pp. 433-441
2. D.C. Brown: Close range camera calibration. Photogrammetric Engineering, 37(8): pp.855-866, 1971
3. D.C. Brown: Decentering distortion of lenses. Photogrammetric Engineering,32(3), 1966, pp. 444-462
4. F. Devernay, O. Faugeras: Straight Lines Have to Be Straight: Automatic Calibration and Removal of Distortion from Scenes of Structured Environments, Machine Vision and Applications, 2001, vol.1, pp.14-24
5. S. B. Kang: Radial distortion snakes, IAPR Workshop on MVA, 2000, pp. 603-606
6. C. F. Marcus: A note on fitting great circles by least squares, Communications of the ACM, 4(11), 1961
7. B. Micusik and T. Pajdla: Estimation of Omnidirectional Camera Model from Epipolar Geometry, CVPR, 2003
8. S. Shah, J. K. Aggarwal: Intrinsic Parameter Calibration Procedure for a (High Distortion) Fish-Eye Lens Camera with Distortion Model and Accuracy Estimation. Pattern Recognition, 1996, vol.29, no.11, pp.1775-1788
9. R. Swaminathan, S.K. Nayar: Non-Metric Calibration of Wide-Angle Lenses and Polycameras. PAMI, 2000, pp. 1172-1178
10. Y. Xiong, K. Turkowski: Creating Image-Based VR Using a Self-Calibrating Fisheye Lens. Proceedings of CVPR, 1997, pp. 237-243
11. X. Ying, Z. Hu: Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model. Proceedings of ECCV, 2004(1), pp. 442-455

Robust Linear Auto-calibration of a Moving Camera from Image Sequences

Thorsten Thormählen, Hellward Broszio, and Patrick Mikulastik

University of Hannover, Information Technology Laboratory,
Schneiderberg 32, 30167 Hannover, Germany
{thormae, broszio, mikulast}@tnt.uni-hannover.de
http://www.digilab.uni-hannover.de

Abstract. A robust linear method for auto-calibration of a moving camera from image sequences is presented. Known techniques for auto-calibration have problems with critical motion sequences or biased estimates. The proposed approach uses known linear equations that are weighted by variable factors. Experiments show, that this modification reduces problems with critical motion sequences and that the estimates are not biased. Therefore, the proposed approach is more robust and achieves a higher estimation accuracy.

1 Introduction

Estimation of camera motion and structure of rigid objects using camera images from multiple views is a common task in computer vision and of interest for many applications. This paper considers the case where the camera performs a translational as well as rotational motion.

For the estimation of the camera motion the real camera is represented by a parametric model, which describes the mapping of the observed three-dimensional rigid objects in the two-dimensional image plane of the camera. The parameters of the camera model can be divided into internal and external camera parameters. External camera parameters describe the position and orientation of the camera in space. Internal camera parameters describe aspects of mapping, e.g. the focal length or the position of the principal point. If the internal camera parameters are known, the camera is calibrated. If the camera is not calibrated, it can be described by the projective camera model. The parameters of the projective camera model are combinations of internal and external camera parameters [1, 2, 3].

In order to estimate the parameters of the projective camera model most approaches establish corresponding feature points in the images. During the estimation of the camera parameters, 3D object points are estimated simultaneously. The resulting reconstruction of projective camera views and object points is determined only up to a global projective transformation. This is sufficient for some applications, for example the synthesis of new views [4]. However, in most applications, the projective reconstruction must be transferred into a metric reconstruction. Therefore, the unknown global projective transformation is reduced to an unknown global metric transformation, which corresponds to a determination of the internal camera parameters and the plane at infinity. Their automatic determination from the parameters of the projective camera is called *auto-calibration*.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 71–80, 2006.

Early publications assumed that the internal camera parameters are constant over the image sequence. In 1992 Maybank and Faugeras [5, 6] used the equations of Kruppa [7]. The method was developed further [8, 9, 10]. In 1997 Triggs [11] presented the *Absolute Dual Quadric* (ADQ), which was later used by Pollefeys et al. [12, 13, 14] for auto-calibration with variable internal camera parameters. An alternative approach first determines the plane at infinity and afterwards the internal camera parameters. The search range for the plane at infinity in the projective space can be limited by the fact that all observed object points must be located in front of the camera [15, 16, 17, 18].

Our approach is a modification of the linear ADQ approach by Pollefeys et al. In [14] Pollefeys et al. weight the linear equations by the reciprocal of the assumed standard deviations of the internal camera parameters. This incorporation of a priori knowledge reduces the problem with critical motion sequences [19, 20, 21]. However, constraining all internal parameters with fixed weights causes biased estimates, even if it is not necessary, e.g. in cases of sequences without critical motion.

In this paper we try to overcome this disadvantage by introducing linear estimation with variable weights instead of fixed weights.

The following Section briefly reviews Pollefeys' approach with fixed weights. In Section 3 the proposed approach with variable weights is presented. Chapter 4 compares results of the different approaches and conclusions are drawn in Section 5.

2 Linear Auto-calibration Using the Absolute Dual Quadric

Starting point of the auto-calibration algorithm is a projective reconstruction with $k = 1 \ldots K$ projective camera views given by the 3×4 camera matrices $\mathtt{A}_k$ and $j = 1 \ldots J$ object points given by the 4-vectors $\mathbf{P}_j$ in homogeneous coordinates.

Auto-calibration determines the projective 4×4 matrix $\mathtt{T}$, that transforms the projective camera $\mathtt{A}_k$ into a metric camera $\mathtt{A}_k^M$:

$$\mathtt{A}_k^M = \mathtt{A}_k \, \mathtt{T} \qquad \forall \quad k \tag{1}$$

and the object points $\mathbf{P}_j$ of the projective reconstruction into metric object points $\mathbf{P}_j^M$:

$$\mathbf{P}_j^M = \mathtt{T} \, \mathbf{P}_j \qquad \forall \quad j. \tag{2}$$

Whereby a metric camera matrix can be factorized as follows:

$$\mathtt{A}^M = \mathtt{K} \, \mathtt{R} \, [\, \mathtt{I} \mid -\mathbf{C} \,]. \tag{3}$$

The 3×3 rotation matrix $\mathtt{R}$ represents the orientation and the 3-vector $\mathbf{C}$ represents the position of the camera. $\mathtt{K}$ is the calibration matrix with

$$\mathtt{K} = \begin{bmatrix} f & s & c_x \\ 0 & r\,f & c_y \\ 0 & 0 & 1 \end{bmatrix} , \tag{4}$$

where f is the focal length, $(c_x, c_y)^\top$ is the principal point offset from the image center, r is the aspect ratio of pixels and s is the skew parameter. The skew s of a real camera is known to be zero. Furthermore, we assume, that the aspect ratio r is known.

In order to determine $\mathtt{T}$, the ADQ $\mathtt{Q}^*_\infty$ is estimated by solving the following auto-calibration equation for all camera views k:

$$\mathtt{A}_k\ \mathtt{Q}^*_\infty\ \mathtt{A}_k^\top \sim \mathtt{K}_k\ \mathtt{K}_k^\top = \omega_k^* \qquad \forall \quad k, \tag{5}$$

where $\mathtt{Q}^*_\infty$ is a 4×4 matrix with rank 3. The 3×3 matrix ω_k^* represents the dual image of the absolute conic (see [2] for details).

In the first step of the linear estimation algorithm the camera matrices are normalized

$$\mathtt{A}'_k = \mathtt{K}_N^{-1}\mathtt{A}_k \tag{6}$$

with

$$\mathtt{K}_N = \mathrm{diag}\left[N_x + N_y,\ \ \frac{1}{r}(N_x + Ny),\ \ 1\right], \tag{7}$$

where N_x is the width and N_y is the height of the camera image. Consequently, the normalized auto-calibration equation is

$$\mathtt{A}'_k\ \mathtt{Q}^*_\infty\ \mathtt{A}_k'^\top \sim \mathtt{K}_N^{-1}\mathtt{K}_k\ \mathtt{K}_k^\top\mathtt{K}_N^{-\top} = \omega_k'^* \qquad \forall \quad k. \tag{8}$$

After the normalization step the focal length of the normalized camera is $f' \approx 1$ and the principal point offset $(c'_x,\, c'_y)^\top \approx (0,\, 0)^\top$. Pollefeys assumes the standard deviations of the unknown normalized parameters to

$$f' \approx 1 \pm 3 \tag{9}$$

$$c'_x \approx 0 \pm 0.1 \tag{10}$$

$$c'_y \approx 0 \pm 0.1. \tag{11}$$

From Eq. (8) follows:

$$\omega_k'^* = \begin{bmatrix} f'^2 + c_x'^2 & c'_x\, c'_y & c'_x \\ c'_x\, c'_y & f'^2 + c_x'^2 & c'_y \\ c'_x & c'_y & 1 \end{bmatrix} \approx \begin{bmatrix} 1 \pm 9.01 & \pm 0.01 & \pm 0.1 \\ \pm 0.01 & 1 \pm 9.01 & \pm 0.1 \\ \pm 0.1 & \pm 0.1 & 1 \end{bmatrix}. \tag{12}$$

The symmetrical 4×4 matrix of the ADQ can be parameterized with 10 elements:

$$\mathtt{Q}^*_\infty = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \\ q_2 & q_5 & q_6 & q_7 \\ q_3 & q_6 & q_8 & q_9 \\ q_4 & q_7 & q_9 & q_{10} \end{bmatrix}. \tag{13}$$

In order to estimate the elements of $\mathtt{Q}^*$, for each camera view 6 linear equations from the following 6 conditions can be derived. Each linear equation is weighted according to its assumed standard deviations from Eq. (12):

$$\omega'^{*}_{12} = 0 \Rightarrow \tfrac{1}{0.01}\,(\mathbf{a}'_1\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_2) = 0 \tag{14}$$
$$\omega'^{*}_{13} = 0 \Rightarrow \tfrac{1}{0.1}\,(\mathbf{a}'_1\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_3) = 0 \tag{15}$$
$$\omega'^{*}_{23} = 0 \Rightarrow \tfrac{1}{0.1}\,(\mathbf{a}'_2\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_3) = 0 \tag{16}$$
$$\omega'^{*}_{11} = \omega'^{*}_{22} \Rightarrow \tfrac{1}{0.2}\,(\mathbf{a}'_1\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_1 - \mathbf{a}'_2\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_2) = 0 \tag{17}$$
$$\omega'^{*}_{11} = \omega'^{*}_{33} \Rightarrow \tfrac{1}{9.01}\,(\mathbf{a}'_1\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_1 - \mathbf{a}'_3\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_3) = 0 \tag{18}$$
$$\omega'^{*}_{22} = \omega'^{*}_{33} \Rightarrow \tfrac{1}{9.01}\,(\mathbf{a}'_2\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_2 - \mathbf{a}'_3\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_3) = 0\,, \tag{19}$$

where $\mathbf{a}'_1, \mathbf{a}'_2, \mathbf{a}'_3$ are the rows of the normalized camera matrix $\mathtt{A}'$.

If the number of camera views is at least 3, an over-determined linear set of equations for the elements of $\mathtt{Q}^*_\infty$ can be generated, which is solved by singular value decomposition [22]. The searched transformation $\mathtt{T}$ can be determined by a singular value decomposition of $\mathtt{Q}^*_\infty$:

$$\begin{aligned} \mathtt{Q}^*_\infty &= \mathtt{U}\ \mathrm{diag}[w_1,\ w_2,\ w_3,\ w_4]\ \mathtt{V}^\top \\ \mathtt{T} &= [\mathtt{U}_3\ \mathrm{diag}[\sqrt{w_1},\ \sqrt{w_2},\ \sqrt{w_3}]\ |\ (0,0,0,1)^\top]\,, \end{aligned} \tag{20}$$

where the columns of the 4×3 matrix $\mathtt{U}_3$ are those three columns of the 4×4 matrix $\mathtt{U}$, which do not correspond to the smallest singular value w_4.

3 Linear Auto-calibration with Variable Weights

In order to improve the above algorithm, we propose to use variable weights for Eqs. (18) and (19) instead of the fixed values:

$$\text{Eq. (18)} \quad \Rightarrow \quad \tfrac{1}{\beta}\,(\mathbf{a}'_1\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_1 - \mathbf{a}'_3\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_3) = 0 \tag{21}$$
$$\text{Eq. (19)} \quad \Rightarrow \quad \tfrac{1}{\beta}\,(\mathbf{a}'_2\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_2 - \mathbf{a}'_3\ \mathtt{Q}^*_\infty\ \mathbf{a}'^{\top}_3) = 0 \tag{22}$$

with

$$\beta \quad = \quad 0.1\, e^{(0.3\, n)}. \tag{23}$$

The modified linear algorithm is executed $N = 50$ times with $n = 0$ to $(N - 1)$.

By altering β exponentially, it is possible to cover a wide range of weights. If $n = 0 \ \Rightarrow\ \beta = 0.1$, and therefore Eqs. (21) and (22) are considered approximately as much as Eqs. (15)-(17) in the linear equation set. If $n = 49 \ \Rightarrow\ \beta = 242174.76$, and the influence of Eqs. (21) and (22) is negligible.

Changing the weight of Eqs. (18) and (19) correspond to changing the assumed standard deviation of the normalized focal length f' in Eq. (9). Another possibility would be to alter the weights of Eqs. (14) to (17), which would correspond to a change of the assumed standard deviation of the principal point offset in Eqs. (9) and (10). However, this would yield the same results, because the result of the equation set is not changed by a global scale and therefore only the ratio of the assumed standard deviations is important.

Since the modified linear algorithm is executed 50 times with different weights, there are 50 possible solutions for T. Each solution is evaluated by the non-linear cost function, which is proposed by Nistér [18]:

$$\phi = \sum_k \frac{s(\mathtt{A}_k\mathtt{T})^2 + c_x(\mathtt{A}_k\mathtt{T})^2 + c_y(\mathtt{A}_k\mathtt{T})^2 + (r(\mathtt{A}_k\mathtt{T}) - r)^2}{f(\mathtt{A}_k\,\mathtt{T})^2} \tag{24}$$

where the functions $s(.)$, $c_x(.)$, $c_y(.)$, $r(.)$ and $f(.)$ extract respectively the parameters skew, principal point offset in x- and y-direction, pixel aspect ratio and focal length from the camera matrix by QR-decomposition [22]. Finally, the solution with the smallest cost ϕ is selected.

4 Results

4.1 Synthetic Data Experiments

In this subsection two experiments with synthetically generated input data are presented. The first experiment simulates a critical camera motion, that is close to a degenerated case, and the second experiment simulates a non-critical camera motion.

For each experiment 500 synthetic test sequences with random scenes are generated. The random scenes consist of 6000 3D object points, which have a distance from the camera between 36 and 72 mm. Each test sequence consists of 10 images. Approximately 160 to 170 of the object points are visible in each camera image. The errors in the positions of the generated 2D image feature points obey an isotropic Gaussian distribution with standard deviation σ. The camera image has 720×576 pixel and a physical size of 7.68×5.76 mm, thus the pixel aspect ratio is 1.06667. The focal length is 10.74 mm. Principle point offset and skew of the camera are zero. All intrinsic camera parameters are kept constant over the sequence.

In experiment 1 translation and rotation between two successive views are very small (see Tab. 1).

Table 1. Camera motion between two successive views for experiment 1 and 2

	Translation [mm]	Rotation [deg]
Exp. 1	X = 0.25	pan = -0.05
	Y = 0.0	tilt = -0.075
	Z = 0.05	roll = 0.005
Exp. 2	X = 2.0	pan = -2.0
	Y = 0.0	tilt = -0.5
	Z = 1.0	roll = 0.05

Fig. 1 shows the results of experiment 1 for five different standard deviations σ of the position errors of generated 2D feature points. The mean and the standard deviation of the estimation results for all intrinsic camera parameters are plotted. Three different approaches for linear auto-calibration using the ADQ are compared: (#1) The approach with fixed weights described in Sec. 2, (#2) the classical approach that does not weight its linear equations and builds its equation set only with Eqs. 14-17, and (#3) the proposed approach with variable weights.

From Fig. 1a the disadvantage of the approach (#1) with fixed weights is evident. The estimation results for the focal length are pulled to the assumed value of

$$\begin{aligned} N_x + N_y &= (720 + 576)\,\text{pixel} \\ &= (7.68 + 5.76)\,\text{mm} = 13.44\,\text{mm} \end{aligned} \tag{25}$$

by Eqs. (18) and (19). Consequentially, the estimation is biased.

On the other hand, if σ is high, approach (#1) gives much better results for all intrinsic parameters (Fig. 1a-e) than approach (#2). The higher robustness against critical camera motions of approach (#1) is due to the additional equations 18 and 19, which are not used by approach (#2). The proposed approach (#3) with variable weights always performs best.

In experiment 2 translation and rotation between two successive views is large and not close to a critical camera motion (see Tab. 1). Thus, the classical approach (#2) gives good estimation results (Fig. 2). Therefore, the biased estimation results of approach (#1) are unnecessary in this case. In contrast, the estimation results of the proposed approach (#3) with variable weights are as good as the results of approach (#2).

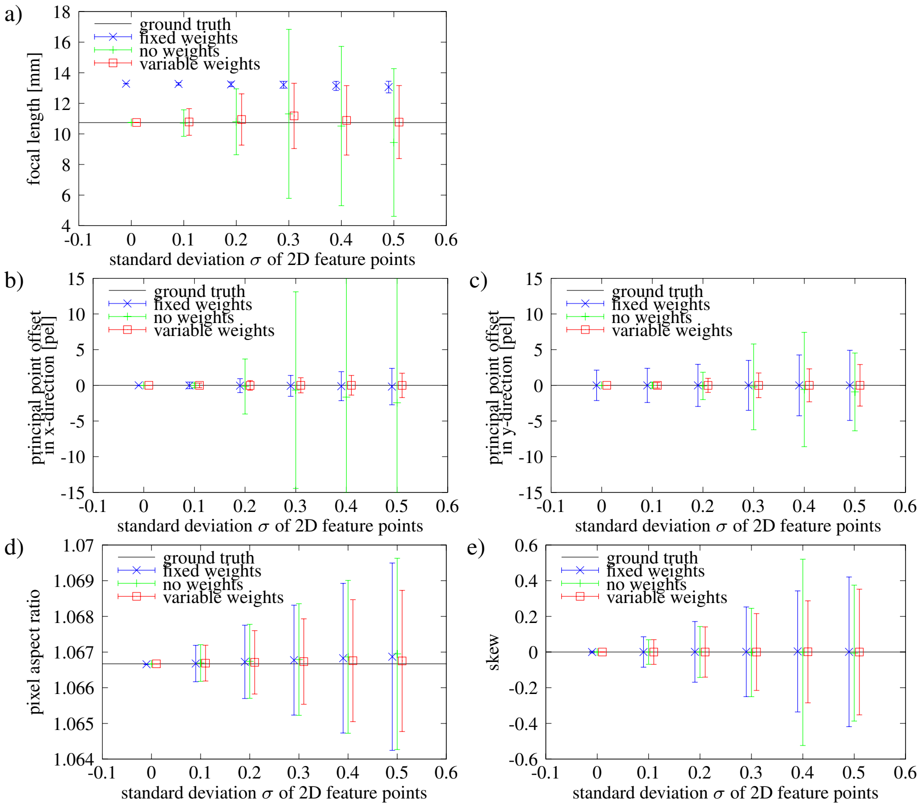


Fig. 1. Results of experiment 1 (critical camera motion): Fig. a)-e) show the ground truth and estimation results of the different approaches for all 5 intrinsic camera parameters over 5 different standard deviations of the position errors of generated 2D feature points. The small symbols mark the mean and the errorbars indicate the standard deviation of the estimation results over 500 random trials.

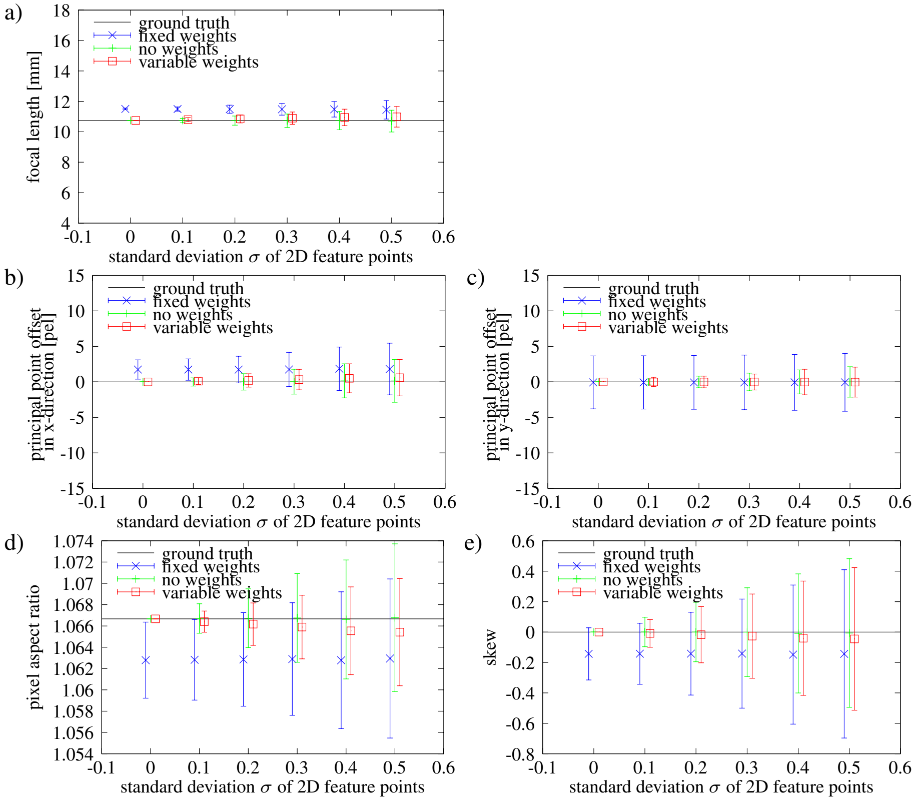


Fig. 2. Results of experiment 2 (non-critical camera motion): Fig. a)-e) show the ground truth and estimation results of the different approaches for all 5 intrinsic camera parameters over 5 different standard deviations of the position errors of generated 2D feature points. The small symbols mark the mean and the errorbars indicate the standard deviation of the estimation results over 500 random trials.

4.2 Natural Image Sequences

The proposed linear auto-calibration approach has also demonstrated to work well on natural image sequences taken by a moving camera. Results of augmented image sequences that have been calibrated using the technique described in this paper are illustrated in Fig. 3. Videos of these augmented image sequences and executables of our non-commercial camera tracker can be found on our website[1].

5 Conclusion

As shown by the experiments the proposed linear auto-calibration approach has nearly no estimation bias and reduces the problem with critical motion sequences. Therefore, it is more robust and achieves an overall higher estimation accuracy than existing approaches.

[1] http://www.digilab.uni-hannover.de

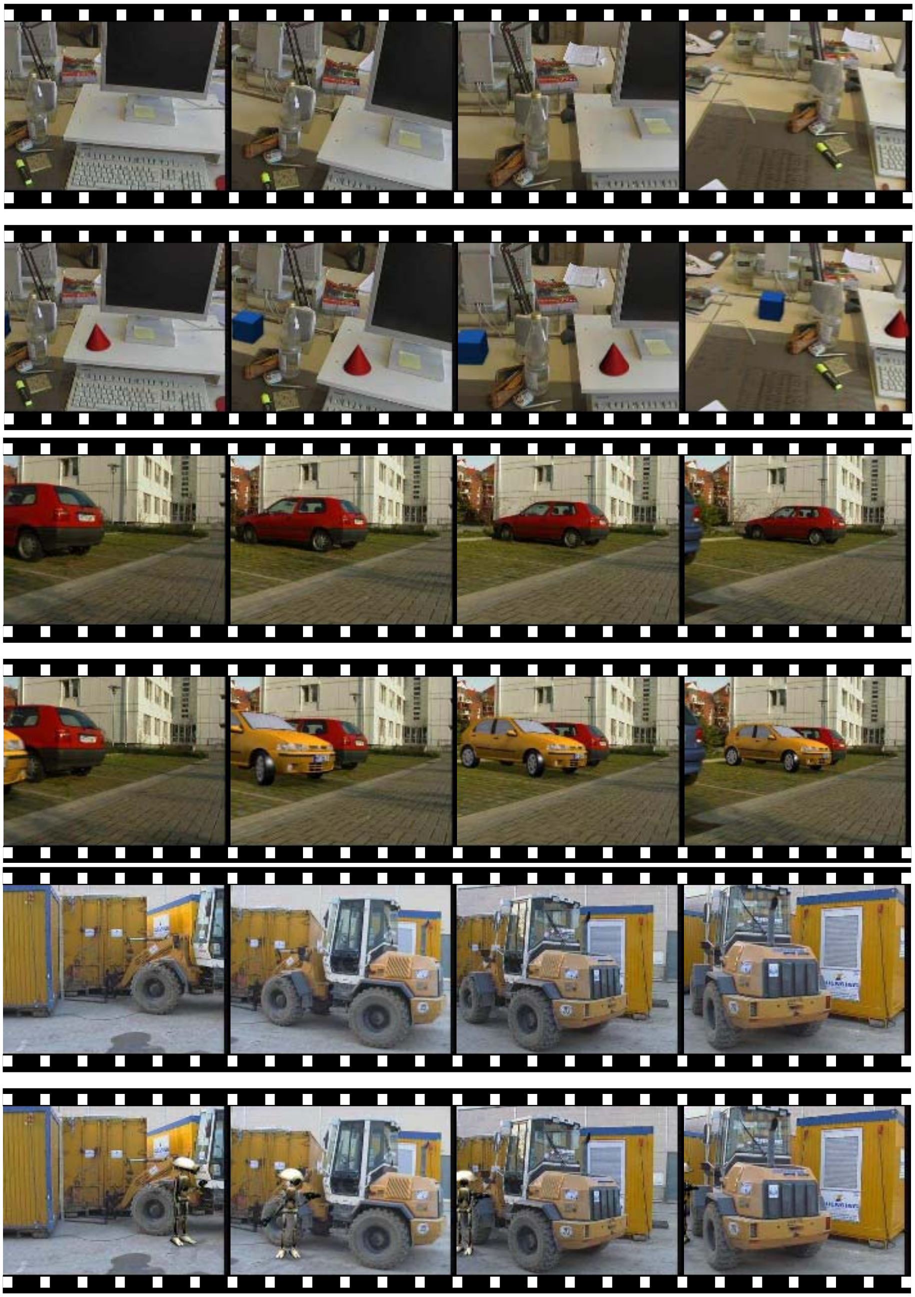


Fig. 3. Examples of augmented image sequences

A slight disadvantage of the proposed approach is its approximately $N = 50$ times higher computational effort. In practice however, this causes no problem, because the computational effort of the linear auto-calibration is small compared to the effort for feature tracking, outlier elimination and estimation of a projective reconstruction. Nevertheless, in future work, it can be tried to reduce N, e.g. by a more explicit detection of critical camera motions.

References

1. Faugeras, O.: Three-Dimensional Computer Vision. MIT Press (1993)
2. Hartley, R.I., Zisserman, A.: Multiple View Geometry. Cambridge University Press (2000)
3. Faugeras, O., Luong, Q.T.: The Geometry of Multiple Images : The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications. MIT Press (2001)
4. Chen, Q.: Multi-view Image-Based Rendering and Modeling. Dissertation, University of Southern California (2000)
5. Maybank, S., Faugeras, O.: A theory of self-calibration of a moving camera. International Journal of Computer Vision **8** (1992) 123–151
6. Faugeras, O., Luong, Q.T., Maybank, S.J.: Camera self-calibration: Theory and experiments. In: ECCV. Volume 558 of Lecture Notes in Computer Science. (1992) 321–334
7. Kruppa, E.: Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung. Sitz-Ber. Akad. Wiss., Wien, Math. Naturw. Abt. IIa. **122** (1913) 1939–1948
8. Heyden, A., Åström, K.: Euclidean reconstruction from constant intrinsic parameters. In: International Conference on Pattern Recognition. Volume 1. (1996) 339–343
9. Zeller, C.: Calibration projective, affine et euclidienne en vision par ordinateur et application a la perception tridimensionnelle. Dissertation, École Polytechnique (1996)
10. Luong, Q.T., Faugeras, O.D.: Self-calibration of a moving camera from point correspondences and fundamental matrices. International Journal of Computer Vision **22** (1997) 261–289
11. Triggs, B.: Autocalibration and the absolute quadric. In: CVPR. (1997) 609–614
12. Pollefeys, M., Koch, R., Gool, L.V.: Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In: ICCV. (1998) 90–95
13. Pollefeys, M., Gool, L.V., Vergauwen, M., Cornelis, K., Verbiest, F., Tops, J.: Video-to-3d. In: Proceedings of Photogrammetric Computer Vision 2002 (ISPRS Commission III Symposium), International Archive of Photogrammetry and Remote Sensing. Volume 34. (2002) 252–258
14. Pollefeys, M., Gool, L.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. International Journal of Computer Vision **59** (2004) 207–232
15. Hartley, R.I.: Cheirality invariants. In: DARPA Image Understanding Workshop. (1993) 745–753
16. Hartley, R., Hayman, E., Agapito, L., Reid, I.: Camera calibration and the search for infinity. In: ICCV. (1999) 510–517
17. Pollefeys, M., Koch, R., Gool, L.V.: A stratified approach to metric self-calibration. In: CVPR. (1997) 407–412
18. Nistér, D.: Calibration with robust use of cheirality by quasi-affine reconstruction of the set of camera projection centres. In: ICCV. Volume 2. (2001) 116–123

19. Sturm, P.: Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In: CVPR. (1997) 1100–1105
20. Kahl, F., Triggs, B.: Critical motions in euclidean structure from motion. In: CVPR. Volume 2. (1999) 367–372
21. Sturm, P.: A case against kruppa's equations for camera self-calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 1199–1204
22. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in C, 2nd ed. Cambridge Univ. Press (1992)

Frame Rate Stabilization by Variable Resolution Shape Reconstruction for On-Line Free-Viewpoint Video Generation

Rui Nabeshima, Megumu Ueda, Daisaku Arita, and Rin-ichiro Taniguchi

Department of Intelligent Systems, Kyushu University,
6-1, Kasuga-koen, Kasuga, Fukuoka 816–8580, Japan
{nabeshima, ueda, arita, rin}@limu.is.kyushu-u.ac.jp

Abstract. Recently, the number of researches aiming at showing real world objects from arbitrary viewpoint have been steadily growing. The processing method is divided into three stages: 3D shape reconstruction by the visual cone intersection method, conversion of 3D shape representation from a voxel form into a triangular patch form, and coloring triangular patches. If the surface area of the object becomes larger, the frame rate decreases since the processing time of the conversion and coloring depends on the number of triangular patches. Stability of the frame rate is essential for on-line distribution of a free-viewpoint video. To solve this problem, we propose a new method which accommodates the space resolution during the 3D shape reconstruction step, thus stabilizing the number of triangular patches and the frame rate. This is achieved by raising the space resolution step by step and stopping the process on a time criteria. The reconstruction is done by using an octree-based visual cone intersection method. Experimental results show that this method makes the frame rate more stable.

1 Introduction

Currently, radio, television, etc. are used as telepresence. However, media using 3D information are more intuitive than those using 1D or 2D information like them because the world in which people live is 3D space. So, several researches have been done for generating the free-viewpoint video which shows objects in the real world from an arbitrary viewpoint using multiple cameras since Kanade et al. [1] had proposed the concept of ”Virtualized Reality” [2][3]. However, they cannot generate the free-viewpoint video in real-time.

We are researching on on-line generation of free-viewpoint video[4], whose process consists of three stages:

1. reconstructing 3D shapes (voxel form) by the visual cone intersection method[5],
2. converting the voxel form of 3D shapes to the triangular patch form of them, and
3. coloring triangular patches.

As shown in the first stage, we reconstruct 3D shapes of objects explicitly. While, Matusik et al.[6] have proposed the image-based visual hull technique, which can generate a free-viewpoint video in real-time. This mathed generates a virtual viewpoint video by projecting rays from the viewpoint on image planes of cameras. This means that the

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 81–90, 2006.

methed does not reconstruct 3D shapes explicitly but implicitly and the amount of computation is smaller than explicitly reconstructing methods. However, since the method directly generates a virtual viewpoint video, the amount of computation is increased relatively to the number of virtual viewpoints. So, it is impossible for the method to deliver free-viewpoint videos to multiple viewers, each of whom can control his/her own virtual viewpoint. On the oter hand, our method can broadcast the triangular patch form of objects with color information, or a CG model, to all viewers and generate free-viewpoint videos on thire own terminals. This means that the amount of comuptation does not depend on the number of viewers.

However, there is a problem. When the surface area of objects becomes larger, the frame rate becomes lower since processing time of the second stage and the third one depends on the number of triangular patches. Stability of the frame rate is very important for on-line distribution of free-viewpoint videos since large jitters of on-line distribution requires longer queues for data transmission.

In this paper, we propose a new method to stabilize the number of triangular patches and, or the frame rate, by dynamically changing space resolution of 3D shape reconstruction. This is realized by using an octree-based visual cone intersection method[7] and raising its resolution step by step until the time allowed for one frame is over.

2 Free-Viewpoint Video

In this section, we show the configuration of the conventional free-viewpoint video system.

Generating a free-viewpoint video is quite time consuming. Therefore, we use a PC-cluster and RPV [8], which is a programming environment for a real-time image processing on a distributed parallel computer (such as a PC-cluster). Multiple cameras synchronized by an external trigger generator are placed in a convergent setup around the center of the scene. Each camera is connected to a PC.

First, object silhouettes are extracted from video frames captured by a camera via background subtraction and noise reduction. For each object silhouette, a visual cone, which is defined as the cone whose apex is the viewpoint and whose cross section coincides with the silhouette, is obtained. Visual cones are represented in terms of voxel space. Then, visual cones from multiple viewpoints are gathered and intersected to construct the shape of the object. Thereafter, 3D shape representation is converted from a voxel form to a triangular patch form by the discrete marching cubes method [9]. Then, visible vertexes of triangular patches are colored based on one camera image. Then, color information of all cameras are integrated. Finally, free-view point images are obtained with color information from a virtual viewpoint directed by the user.

3 Visual Cone Intersection Using Octree

Increasing the space resolution step by step by the visual cone intersection method which we used is difficult. Indeed this method must scan all voxels in the finest space resolution. Therefore, we use an octree-based visual cone intersection method proposed by Sato et al.

3.1 Octree

An *octree* is a tree to index three dimensions. Each node has either eight children or no children. This means that one voxel is divided into eight voxels when the space resolution is increased.

3.2 Pass Algorithm

Each voxel is classified into three types as follows.

White. The eight vertices are projected on the silhouettes
Black. The eight vertices are projected on the background
Gray. The other voxels (borders of silhouettes)

Voxels of a pre-defined size called initial voxels are classified into the above three voxel types. After that, only gray voxels are subdivided. Such classification and subdivision procedure is recursively executed until a pre-defined finest space resolution. This subdivision procedure from an also pre-defined coarse space resolution to the finest one is named *pass* and the algorithm is named *pass algorithm*.

3.3 Multi-pass Subdivision Algorithm

Since the intersection test described above is very simple, it sometimes ends in failure. Fig. 1 shows an example of that case. In the example, a voxel is projected on a narrow tip of a silhouette that avoids its vertices. This voxel must be classified as gray. Nevertheless, it is classified as black. This voxel is named *failed voxel*. Then, a Multi-Pass Subdivision Algorithm (see Fig. 2) is proposed to solve this problem. In Step1, the pass algorithm is applied to initial voxels. Afterward, failed voxels are detected in Step2. Failed voxels are subdivided and the pass algorithm is applied to the voxels obtained in Step3. Failed voxels are detected once again in Step2. As long as failed voxels are detected, Step2 and Step3 are repeated. If no failed voxels are detected, the process ends.

The way to detect failed voxels is as follows. We connect neighboring gray voxels in the finest space resolution, and thus we get the surface of a reconstructed geometry. If there is a white voxel neighboring a black voxel, the surface is not closed. Therefore, we regard the both voxels as failed voxels. This also means that there cannot be failed voxels in the finest space resolution. As a result, failed voxels can always be divided.

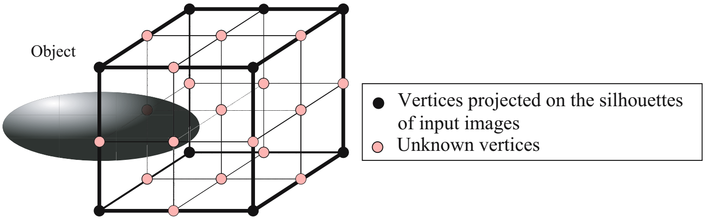


Fig. 1. Failed Voxel (Case Example)

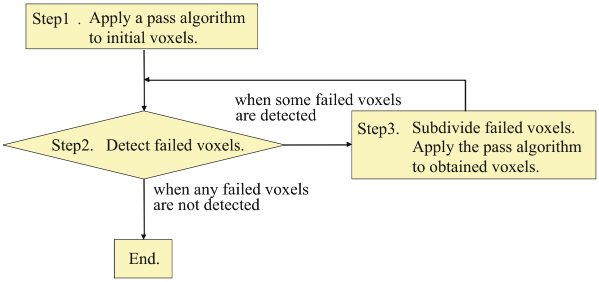

Step1 . Apply a pass algorithm to initial voxels.
when some failed voxels are detected
Step2. Detect failed voxels.
Step3. Subdivide failed voxels. Apply the pass algorithm to obtained voxels.
when any failed voxels are not detected
End.

Fig. 2. Multi-pass Subdivision Algorithm

4 Frame Rate Stabilization by Variable Resolution Shape Reconstruction

4.1 Variable Resolution Shape Reconstruction

We use the octree-based visual cone intersection method to reconstruct the 3D shape in variable space resolution. Using this method, even if the process of 3D shape reconstruction does not reach leaf nodes of the octree, we can reconstruct the shape in lower space resolution using the information of the ancestor nodes. Then, we can stop the process and stabilize the frame rate.

The process of 3D shape reconstruction is shown in Fig. 3. The procedure from Step2 to Step4 is recursively executed until a certain time is over. When the time passes, the process stops at the end of either Step2 or Step3. We call the resolution in which the process is stopped *cutoff resolution*.

Step1. Apply the multi-pass subdivision algorithm in a certain cutoff resolution
Step2. Send leaf nodes in the current cutoff resolusion
Step3. Subdivide gray voxels and failed voxels in the higher cutoff resolution which cannot be subdivided in the previous subdivision
Step4. Apply the multi-pass subdivision algorithm

In Step1, when a node is judged to be black, the descendants of the node are considered as black nodes. Likewise, when a node is judged to be white, the descendants of the node are also judged to be white. Meanwhile, when a node, which cannot be divided in the cutoff resolution, is judged to be gray, the descendants of the node are judged to be gray.

In Step2, the information, white or black, of each leaf nodes is output. A gray voxel is output as a white voxel. However, when a gray voxel in the finest space resolution, the information of its vertices is output.

In Step1 or Step4, when there are failed voxels which cannot be divided in the current cutoff resolution but can be divided in the next higher cutoff resolution in Step3, they are retained and subdivided in the following Step3.

4.2 Detection of Failed Voxels

The first time we detect failed voxels, i.e. in the coarsest cutoff resolution, we scan all white voxels. For each white voxel, we detect black voxels which are minimum voxels

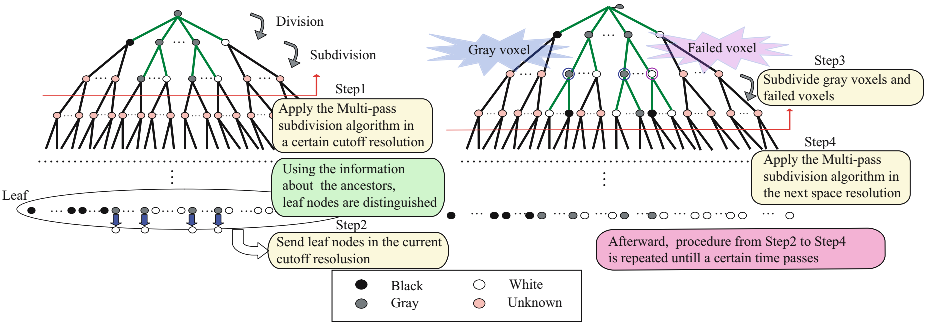

Division
Subdivision
Step1
Apply the Multi-pass subdivision algorithm in a certain cutoff resolution
Using the information about the ancestors, leaf nodes are distinguished
Leaf
Step2
Send leaf nodes in the current cutoff resolution
Gray voxel
Failed voxel
Step3
Subdivide gray voxels and failed voxels
Step4
Apply the Multi-pass subdivision algorithm in the next space resolution
Afterward, procedure from Step2 to Step4 is repeated untill a certain time passes
● Black
● Gray
○ White
● Unknown

Fig. 3. Variable Resolution Shape Reconstruction

in the current cutoff resolution and are neighboring the white voxel. When we detect such a black voxel, we detect a black node which can be divided by tracing a link from the voxel to the parent node. If there is a node which can be divided, we regard the node as a failed voxel and divide it. In addition, if the white voxel can be divided, we also regard the node as failed voxel and divide it.

On the other hand, for the second time or later, we scan such white voxels and black voxels that are obtained by dividing gray voxels and failed voxels on the previous process, assuming failed voxels appeared at the time.

4.3 System Configuration

We obtained visual cones from each camera and intersected them in the conventional system. But we obtain visual cones from each three cameras and intersect them. By doing so, the surface area diminishes and the number of gray voxels decreases. It increases the number of voxels which do not have to be divided, and thus the processing time decreases.

The system configuration which we propose is as following. Processes are distributed to PCs shown in Fig. 4 and executed in pipeline parallel.

Node-A: Each node-A extracts object silhouettes from video frames captured by a camera by background subtraction and noise reduction, and sends the silhouette image to a node-B and a node-D. Each node-A sends a binary format silhouette image to a node-B to reduce the amount of data since each node-B uses the image to reconstruct a visual cone. On the other hand, each node-A sends a RGB silhouette image to a node-D since each node-D uses the image to color a visual hull.

Each node-A’ does not works for model coloring but only for shape model reconstruction. Indeed model coloring needs large processing time, and it becomes difficult to use many cameras. The node-A’s do not send images to node-Ds.

Node-B: Each node-B constructs a visual hull. When a-node-B refers vertices for voxel classification, it projects vertices onto the silhouette images from three viewpoints. Each vertex is regarded as occupied only when all projected point on each image are occupied. Each node-B sends the node-C a visual hull and information about the cutoff resolution one time or more.

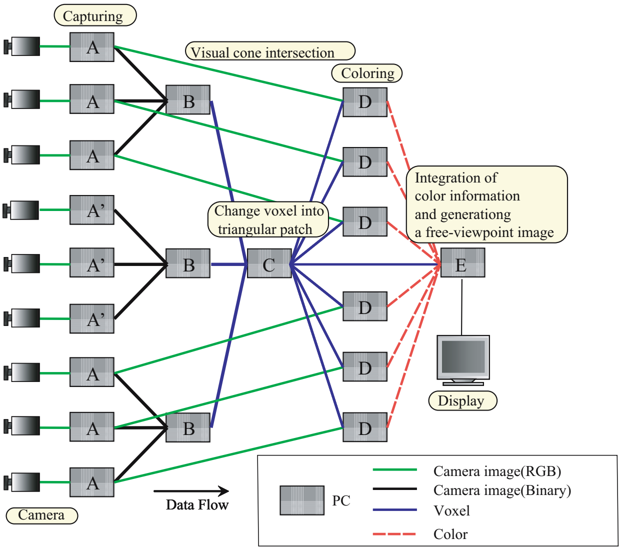


Fig. 4. System Configuration

Node-C: Each node-C intersects the visual hull every time it receives one from node-B. If the cutoff resolutions of all the visual hulls are not same, lower cutoff resolutions are adjusted to the highest cutoff resolution. Furthermore, the node-C converts the final shape model represented in terms of voxel space into triangular patches by the discrete marching cubes method. Then, the node-C sends the voxel space and its corresponding marching cube patterns to node-D and node-E since the data size of both voxel space and its marching cube patterns is smaller than that of all triangular patches.

Node-D and Node-E are the same as conventional system.

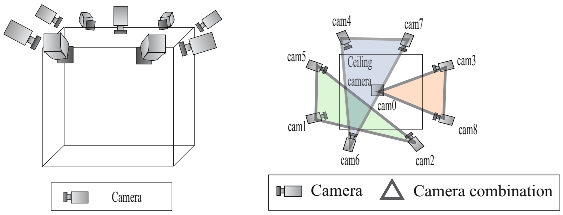
Node-D: First each node-D transforms the shape model represented in terms of a voxel space into triangular patches by using patterns sent from the node-C. Then, each node-D colors visible vertexes of the shape model based on one camera image. Finally, each node-D sends color information of all the vertexes of the shape model to the node-E.

Node-E: First, the node-E receives the position of the virtual viewpoint directed by the user. Then the shape model transforms into triangular patches in the same way as a node-D. Finally, the node-E integrates color information of all cameras and generates an image from the directed viewpoint.

5 Experiments

5.1 Experimental Environments

Using the system we proposed, we generate a free-viewpoint video in real-time to evaluate the processing time and the quality of generated images. We have used nine



(a) From a slanting top

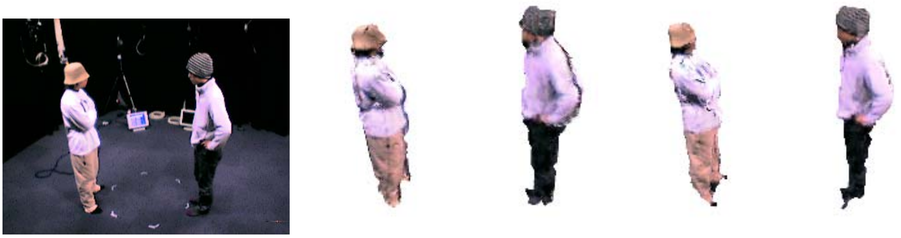
(b) From a top

Fig. 5. Camera arrangement and combination

IEEE-based cameras at 320 × 240 pixel resolution, and 20 PCs (six node-As, three node-A's, three node-Bs, one node-C, six node-Ds, one node-E), each of which has an Intel Pentium4 (3GHz), 1GB memory and NVidia GeForce FX. PCs are connected with each other by Myrinet, a giga-bit network. All the cameras are calibrated in advance by Tsai's method [10]. The camera arrangement and combination of visual cone intersection in node-Bs are shown in Fig. 5. Maximal space resolution is $128 \times 128 \times 128$ and the size of a minimum voxel is $2cm$. The depth of the octree is five and the space resolution of initial voxels is $8 \times 8 \times 8$. The cutoff resolution is two level, $64 \times 64 \times 64$ and $128 \times 128 \times 128$. That is to say, we apply multi-pass algorithm until the depth of the octree is four and space resolution is $64 \times 64 \times 64$, and we advance to $128 \times 128 \times 128$ after sending leaf nodes.

5.2 Generated Free-Viewpoint Video

Fig. 6 shows camera images and generated images from a same viewpoint by the proposed variable resolution system and by conventional fixed resolution system. The space resolution of the fixed resolution system is $128 \times 128 \times 128$. In the variable resolution system, the shape reconstruction process is stopped at $64 \times 64 \times 64$ of the cutoff resolution since the total surface area of the objects, or two persons, is too large. Therefore, the object shapes look coarse because of the low space resolution. Moreover, the objects look bigger than those in the fixed resolution system, since we regard the gray

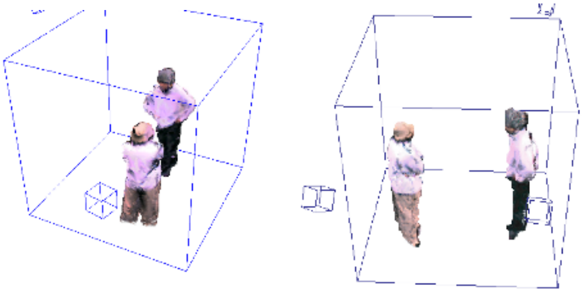


(a) Camera image

(b) Generated image (Variable Resolution System)

(c) Generated image (Fixed Resolution System)

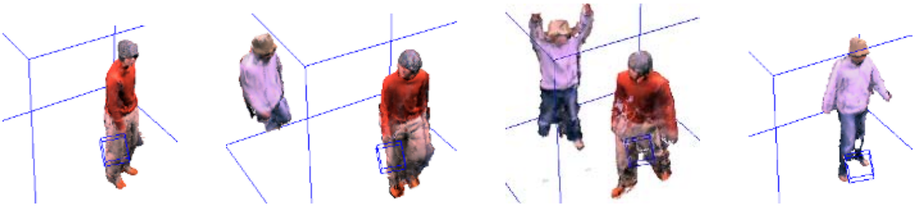
Fig. 6. Camera image and generated image from a same viewpoint



(a) Generated image 1

(b) Generated image 2

Fig. 7. Generated images from virtual viewpoints



(a) Generated image 1

(b) Generated image 2

(c) Generated image 3

(d) Generated image 4

Fig. 8. Generated images in case that the number of persons varies

voxels in the reconstructed surface as white voxels. As a result, the surface of object is not colored correctly.

Fig. 7 shows generated images from virtual viewpoints by the variable resolution system. Small cubes in the images represent real camera positions. Small cubes represent camera position. The process is stopped at 64 × 64 × 64 of the space resolution.

Fig. 8 shows generated image by the variable resolution system in case that the number of persons varies. When there are two persons, the process is stopped at 64 × 64 × 64 of the space resolution. Otherwise, there is enough time to reach 128 × 128 × 128 of the space resolution. This example shows the space resolution is changed according to the objects.

5.3 Processing Time

Not only the average of frame rate but also the variance are important for on-line free-viewpoint video distribution. Indeed, should the variance get higher, the communication buffer size will have increase as well, thus also increasing the delay time.

Fig. 9 shows frame rate in case the number of persons changes with the fixed resolution system (conventional technique) and with the variable resolution system (proposed technique). When the number of people changes from one to two, the frame rate of conventional system decreases significantly. On the other hand, the frame rate with pro-

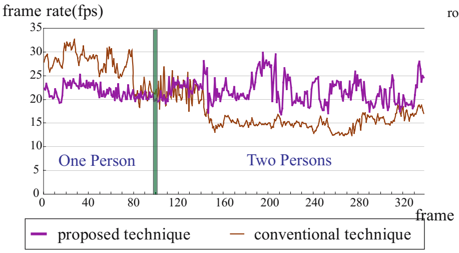


Fig. 9. Frame rate

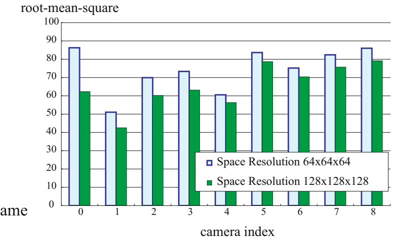


Fig. 10. Error of coloring

posed technique keeps 20fps independent of objects. Yet the frame rate with proposed technique has a small fluctuation. Indeed, stopping the process or not is judged after step2 or step3 (in section4) Stopping the process does not always happen fixed time.

5.4 Quality of Generated Image

When the process is stopped, space resolution is low, and the precision falls. We evaluate how different the error of coloring in space resolution $128 \times 128 \times 128$ and $64 \times 64 \times 64$ is. Fig. 10 shows the sum of root mean square error between a camera image and a generated image with the same viewpoint. The error in lower space resolution increase a little, but the increasing amount is much smaller than the error in space resolution $128 \times 128 \times 128$.

6 Conclusion

In this paper, we propose frame rate stabilization by variable resolution shape reconstruction for on-line free-viewpoint video generation. And we make some experiments to show the frame rate stabilization independent of the object. Major future works are as follows:

- reduction of the fluctuation in the frame rate
- compression of the amount of data transfer by sending not a voxel space but an octree
- realizing a multi-resolution marching cube method
- compression of color information
- developing an on-line free-viewpoint video distribution system.

References

1. T. Kanade, P. W. Rander, P. J. Narayanan:“ Concepts and early results”, IEEE Workshop on the Representation of Visual Scenes, pp.69–76, Jane 1995.
2. Shohei Nobuhara, Takashi Matsuyama:“ Heterogeneous Deformation Model for 3D Shape and Motion Recovery from Multi-Viewpoint Images”, Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission, pp. 566–573, Thessaloniki Greece, September 2004.

3. J. Carranza, C. Theobalt, M. A. Magnor, H.-P. Seidel:“ Freeviewpoint video of human actors”, in ACM Trans. on Graphics, vol. 22, no. 3, pp.569–577, July 2003.
4. Megumu Ueda, Daisaku Arita, Rin-ichiro Taniguchi:“ Real-Time Free-Viewpoint Video Generation Using Multiple Cameras and a PC-Cluster”, Proc. of Pacific-Rim Conference on Multimedia, pp.418–425, December 2004.
5. W. N. Martin, J. K. Aggarwal:“ Volumetric description of objects from multiple views” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 5, No. 2, pp.150–158, 1983.
6. W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan:“ Image-Based Visual Hulls”, Proc. of SIGGRAPH, pp.369–374, 2000.
7. Hidenori Sato, Hiroto Matsuoka, Akira Onozawa, Hitoshi Kitazawa:“ Image-Based Photorealistic 3D Reconstruction Using Hexagonal Representation”, IPSJ Journal, Vol. 46, No. 2, pp.639–648, 2005.
8. Daisaku Arita, Rin-ichiro Taniguchi:“ RPV-II: A stream-based real-time parallel vision system and its application to real-time volume reconstruction”, in Proc. of Second International Workshop on Computer Vision System, pp.174–189, July 2001.
9. Yukiko Kenmochi, Kazunori Kotani, and Atsushi Imiya:“ Marching cubes method with connectivity”, in Proc. on International Conference on Image Processing, vol. 4, pp.361–365, Oct 1999.
10. R. Y. Tsai:“ A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses”, in IEEE Trans. on Robotics and Automation, vol. 3, no. 4, pp.323–344, 1987.

Vision-Based Posing of 3D Virtual Actors

Ameya S.Vaidya, Appu Shaji, and Sharat Chandran

Computer Science and Engineering Department, I.I.T Bombay,
Mumbai 400076, India
{ameyasv, appu, sharat}@cse.iitb.ac.in

Abstract. Construction of key poses is one of the most tedious and time consuming steps in synthesizing of 3D virtual actors. Recent alternate schemes expect the user to specify two inputs. Along with a neutral 3D reference model, more intuitive 2D inputs such as sketches, photographs or video frames are provided. Using these, of all the possible configurations, the "best" 3D virtual actor is posed

In this paper, we provide a solution to this ill-posed problem. We first give a solution to the problem of finding an approximate view consistent with the 2D sketch. Elements of this rigid-body solution are novel. Next, we provide a new solution to the process of extending or retracting limbs to more accurately suit the sketch. This posing algorithm, is based on a search based scheme inspired by anthropometric evidence. Less *physical work* is required by the actor to reach the desired pose from the base position. We also show that our algorithm converges to an acceptable solution much faster compared to the previous methods.

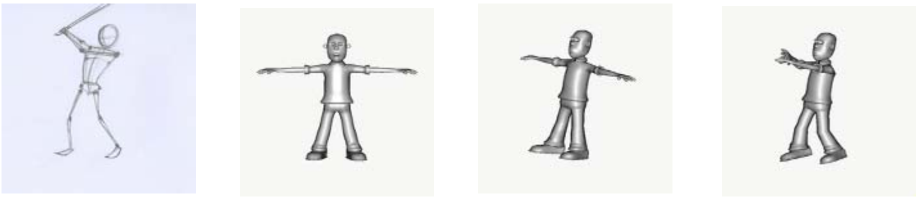
1 Introduction

Consider the following:

- A cricket coach uses an animation system in correcting the flaws in the strokes of one of his current players based on proven, vintage stars of the past. He generates "on the fly" a new three-dimensional (3D) sequence rapidly based on the combination of the 3D model of his current player and past videos at his disposal.
- An occupational therapist takes a scanned picture from her textbook. She overlays correct posture styles for the computer hacker hunched up over his laptop.

Animation systems currently used fall well short of providing the necessary amenities to realize the above. Due to the highly articulate and complex structure of human 3D characters and their respective motions, *posing* them in a 3D world and specifying their motion is by no means a trivial task. Alternative schemes [1] [2] have thus evolved that *compute* a desired pose from a 2D sketch, a photograph, or from a video frame. For simplicity, we assume in the rest of the paper that our inputs are artist drawn sketches, and a reference 3D virtual actor modeled as an articulated skeleton.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 91–100, 2006.



(a) Input Sketch (b) Input 3D Actor (c) Orientation Recovered (d) Pose matches sketch

Fig. 1. A sketch and a 3D actor (top row) is presented to our system. It first (bottom left) re-orients the actor rigidly and then "moves" the limbs to match the sketch. In this example, the positions of all four limbs are computed and the knee adjusted.

1.1 Problem Statement and Contributions

There are two issues in computing a solution as in Fig. 1. First, an approximate viewing direction must be found that orients the 3D actor to match the given sketch. At the end of this step, the 3D actor has rigidly oriented himself to be ready to move to the new configuration in the sketch.

Second, the actor changes relative positions of the "bones" so as to match the sketch.

Note that there are potentially infinite configurations that match the sketch. Multiple positions may be used to construct an animation sequence based on key intermediate poses.

We provide a partially automated solution to this problem. In our scheme, the end user specifies a few correspondences between points on the 2D sketch and points in the 3D reference model. Our system automatically *constructs* the gross orientation. Next, based on this orientation, the system automatically moves the limbs in a non-rigid fashion to match the sketch. We list the features of our work.

1. A more robust (albeit domain-specific) rigid body camera recovery algorithm is presented (see Section 3).
2. For the "most-likely" non-rigid motion, a novel search based scheme inspired by anthropometric[1] evidence is introduced.
3. The notion of a physically based metric to quantify the results is introduced. Compared to existing methods, our scheme requires less physical work to reach the specified sketched pose from the previous position. The application of this to energy efficient robotics is immediate.
4. Compared with prior iterative solutions, our method takes less time and can be done at interactive rates.
5. Since the reconstruction from 2D sketches to 3D poses is not unique, we provide the end user the option to select from multiple solutions. The solutions are returned in an order of "less movement" to "more movement" on the part of the 3D actor.

[1] Anthropometry: measurement and study of the human body and its parts and capacities.

6. The sketches provided by the artist are not expected to be the exact projections of the desired pose. But a loose sense of proportion is expected.

The rest of the paper is organized as follows. After considering related work in Section 2 we give a brief overview of the camera recovery algorithm in Section 3. We present the conceptual and implementation details of our posing algorithm in Section 4. We analyze our results and perform statistical comparisons with earlier methods in Section 5. Final remarks appear at the end.

2 Previous Work

The problem of extracting 3D poses from 2D poses has been tackled in various domains like robotics, CAD/CAM, computer vision, animation and graphics before. A popular approach is to use two or more images from different viewpoints to resolve ambiguity between multiple valid poses.

A technique for reconstructing human body poses from single images with the aid of anthropological data is discussed in [3]. In [4], the authors have outlined a technique that relies on known point correspondences between predefined landmarks on the human body. Most of these are learning based schemes. An alternative strategy, useful for some 3D animation applications, is to use information from "previous" frame(s) when available. See for example, [5].

Another school of thought is to re-structure the problem as an optimization problem [6][7]. The hypothesis behind optimization-based posture prediction is that human motion concerning different tasks is governed by different performance measures. These measures can be aggregated using multi-objective optimization techniques.

Pose recovery techniques close to our stated goals are discussed in [1] and [2]. The method proposed in [2] achieves a good amount of automation but works only with "stick-figures" as 2D input. Also they require the 2D skeleton to be an isomorph of the 3D skeleton, which limits the applicability of the method. This assumption, for example, may not hold true with motion capture tracking data [2]. Our work is essentially patterned around [1]. We re-work the posing scheme so as to make it more robust, faster and closer to actual human motion. Further, our algorithm has the option of returning multiple solutions.

Our posing algorithm is loosely based on Cyclic Coordinate Descent (CCD) method [8] [9]. An excellent introduction of all of the problems and general approaches to Inverse Kinematics is provided in [8].

3 Recovery of Gross Orientation

The key to recover the orientation is to find a "camera" such that when it looks at the 3D shape in the *correct* orientation, the projection of the 3D shape matches the input sketch. Mathematically, the camera is a matrix $\mathbf{P}_{3\times4}$

$$\mathbf{x} = \mathbf{PW} \tag{1}$$

where $\mathbf{W}_{4\times1} = [X\ Y\ Z\ 1]^T$ is an object point and $\mathbf{x}_{3\times1} = [uw\ vw\ w]^T$ is the corresponding image point.

$\mathbf{P}$ can be computed given a set of *user-clicked* point correspondences $(\mathbf{x}_i, \mathbf{W}_i)$, between the image and the reference actor. Normally, at least six point correspondences (in general position) are required for the simplest camera model.

3.1 Our Method

Instead of finding the camera matrix, and then recovering the 3D points, our domain-specific method directly computes the required 3D points. The method requires no more than *five* points such that the clicked points belong to the same object and no four of these points are co-planar.

The clicked sketch points, $\mathbf{x}_i$, are $2D$ projections of the corresponding, currently unknown, $3D$ points $\mathbf{W}^{'}{}_i = [X_i^{'}\ Y_i^{'}\ Z_i^{'}\ 1]$. However, the 3D positions, and hence distances, are known in the reference position. Because the camera recovery phase is a rigid body transformation, the distances between joints in the skeleton is preserved. The unknown points $\mathbf{W}^{'}{}_i$ are computed using this invariant. The details of this step (keeping in mind issues such as the scaling, origin alignment, and the like) are skipped for brevity.

The unknown transformation $\mathbf{T}$ given by $\mathbf{W}^{'}_{\mathbf{4}\times\mathbf{1}} = \mathbf{T}_{\mathbf{4}\times\mathbf{4}}\mathbf{W}_{\mathbf{4}\times\mathbf{1}}$ can now be obtained. Finally, $\mathbf{T}$ is related to the camera matrix by the relation $\mathbf{P}_{\mathbf{3}\times\mathbf{4}} = \mathbf{K}_{\mathbf{3}\times\mathbf{4}}\mathbf{T}_{\mathbf{4}\times\mathbf{4}}$ where $\mathbf{K}_{\mathbf{3}\times\mathbf{4}}$ is the projection matrix. This enables us to find $\mathbf{C}$, the viewing direction, as the right null space of the camera matrix.

4 Non-rigid Posing

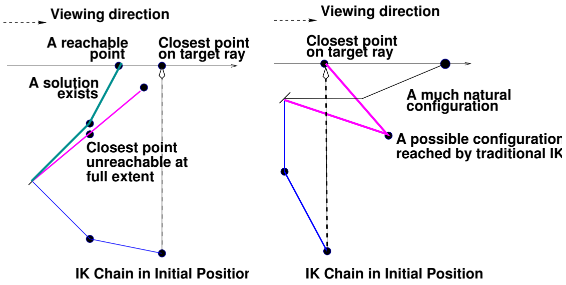
The non-rigid transformation is the next step. The problem is set up as an inverse kinematics problem. At this point, the user clicks corresponding positions of the desired limb, termed as the *end effector*. This is the pair $(\mathbf{e}\ , \mathbf{W})$ on the sketch and the reference character. The system then back-projects the $2D$ sketch position $\mathbf{e}$ to obtain the target ray in 3D space $\mathbf{R}_e = \lambda\mathbf{C} + \mathbf{P}^{-1}\mathbf{e}$ where λ is a real number.

As a minimum, only the position of the end effector is given, and that too in an approximate sense. The true 3D position, and the intermediate joints are not specified by the user. Of course, to construct a more accurate pose, the user may decide to provide the $2D$ positions of the intermediate joints as well, and the system will use this information when available. However we have found that this is rarely necessary. A properly constructed model along with our strategy of returning multiple discrete configurations, as discussed in section 4.2, yields satisfactory poses in most cases.

4.1 Basic Idea

Where on this ray will the actual point lie? This is an important question which drives the quality of the solution. Two choices are considered

– The method proposed in [1] uses the *closest point* on this ray to the *current* end-effector position as the target position and applies traditional inverse


Viewing direction
A reachable point
Closest point on target ray
A solution exists
Closest point unreachable at full extent
IK Chain in Initial Positior
Viewing direction
Closest point on target ray
A much natural configuration
A possible configuratior reached by traditional IK
IK Chain in Initial Position

(a) The closest point is unreachable

(b) Unnatural Pose

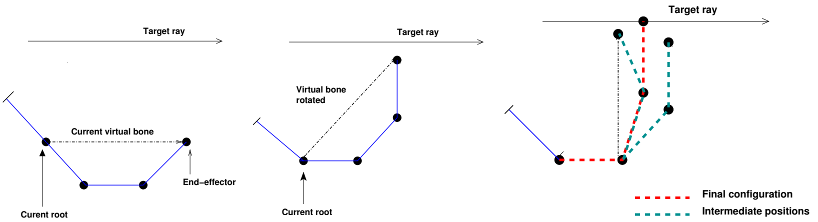
Fig. 2. Problems with closest point assumption

kinematics. However the closest point may be unreachable as seen, for example, in Fig. 2(a) or may lead to unnatural poses as in Fig. 2(b).

- Alternatively, blind Jacobian based inverse kinematics may be used where the first satisfying end point is automatically computed. However, the 3-D reference character has to perform more physical work (Section 5). Besides this method needs more iterations to converge to a solution.

The intuition behind our scheme comes from the way human limbs operate. The human limb motion compromises on various factors like the effort required in planning the motion, the energy expended in executing the motion and stability of the resultant posture. As a result, limb motion occurs by an overall gradual rotation towards the goal along with simultaneous extension or retraction to span the required distance [10].

We mimic the above behavior by using a search based scheme. We use a recursive bi-directional search for the best configuration to reach the target ray starting from the smallest sub-chain. At each step of the recursion, let *current root* be the joint at the base of the current sub-chain. We call the vector from the current root to the end-effector a *virtual bone*. The algorithm orients the current virtual bone by rotating the current root so that the end-effector is closest to



(a) Before orienting current virtual bone

(b) Orienting the current virtual bone

(c) Expanding the sub-chain to reach the ray

Fig. 3. Phases in our algorithm

the target ray. The algorithm then extends or retracts the chain to try and reach the ray. To do this, it recurses with a smaller sub-chain to search for a suitable configuration that places the end-effector on the ray. If successful, the algorithm returns. If we are unsuccessful, but this step reduces the posing error, the resulting configuration is saved before proceeding further. Finally if instead it increases the posing error, the rotation applied to *current root* is undone and the chain is restored to the last saved state[2]. The process is shown in Fig. 3 and the algorithm is given in Algorithm 1. Finally longer sub-chains are considered in Algorithm 2.

Algorithm 1. PoseChain (IN : $start$, IN : end, IN : ray, IN : $thresh$, IN/OUT : $error$) : $Success, Partial$

```
 1: if $error < thresh$ then
 2:    return $Success$
 3: else
 4:    if $start == end$ then
 5:       return $Partial$
 6:    end if
 7: end if
 8: $virtBone \leftarrow (curRoot, end)$
 9: Compute rotation(s) for $virtBone$
10: Select best rotation $bestRot$
11: Save the current value of $curRoot$
12: Apply $bestRot$ to $virtBone$
13: Compute current posing error $locError$
14: if $PoseChain(start.child, end, ray, locError) == Success$ then
15:    return Success
16: end if
17: if $locErr < error$ then
18:    $error = locErr$
19: else
20:    Restore the saved value of $curRoot$ in step 11
21: end if
22: return $Partial$
```

Since `PoseChainTop` considers increasingly longer chains, the time complexity of the algorithm, $T(n)$ in terms of chain length n can be computed by the recurrence

$$T(n) = \sum_{\imath=1}^{\imath=n} T_1(\imath) \tag{2}$$

[2] The desired orientation of the current virtual bone PQ, where P is the current root and Q is the current end-effector, is computed by drawing a sphere centered at the current root and radius equal to the length of the current virtual bone. The closest point to the ray be S. The rotation R is the one which aligns PQ, along PS.

Algorithm 2. PoseChainTop (IN : *start*, IN : *end*, IN : *ray*, IN : *thresh*) : *Success*, *Partial*

```
$error \leftarrow \infty$
for $curRoot = end.parent$ to $start$ do
    if $PoseChain(curRoot, end, ray, thresh, error) == Success$ then
        return $Success$
    end if
end for
return $Partial$
```

where T_1 is the time complexity of the recursive algorithm `PoseChain`. Consider the call to algorithm `PoseChain` with chain size m. To compute $T_1(m)$ we note that it consists of a single recursive call of size $m - 1$ in step 14. All other steps can be done in constant time. Therefore the algorithm `PoseChainTop` is quadratic.

This is much better than a Jacobian based scheme, where at every step a $6 \times n$, matrix must be computed and inverted. Further, in our case the "steps" taken by the IK chain are much larger than the Jacobian scheme. Thus our algorithm executes much faster than the traditional scheme.

`PoseChainTop` is loosely based on CCD [9]. In fact, the **for loop** in the algorithm `PoseChainTop` is a conceptual implementation of CCD. CCD is a linear time algorithm. However the length of an IK chain seldom exceeds 10, hence the execution time of `PoseChainTop` is well within interactive rates. Further, basic CCD suffers from the problem of excessive folding since the search proceeds in only one direction, from the end-effector towards the root of the chain. Once a bone is rotated, the sub-chain rooted at that bone is never considered again. In contrast, our method reconsiders the sub-chain via the recursive call at the end of algorithm 1, thus correcting for the excessive rotation that happens with standard CCD. This also takes care of convergence issues in the presence of joint-constraints[3]. Another advantage we have over CCD is the ability to generate multiple discrete configurations as discussed in section 4.2 below.

4.2 Generating Multiple Configurations

Using a search-based approach allows us to generate multiple discrete configurations using the following strategy. In general, at the point in the search tree where the algorithm returns successfully, the corresponding virtual bone will have two possible orientations—see step 9 of `PoseChain`. In the basic scheme, we select the "best" and discard the other. In the modified version, we cache the second orientation in algorithm 2, subject to joint constraints, before returning. This represents the node in the search tree from which the search must be restarted for other solutions. Now when the user requests another solution,

[3] In our system, joint constraints are modeled in the form of *constraint cones*, enclosing the joint in its parent coordinate frame (see Fig.6).

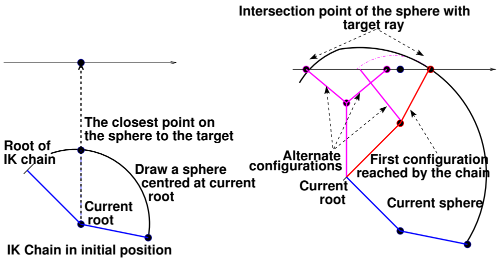
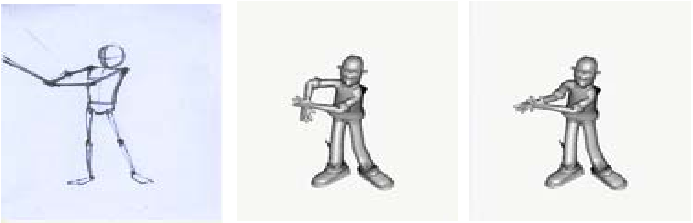

Intersection point of the sphere with target ray
The closest point on the sphere to the target
Root of IK chain
Draw a sphere centred at current root
Current root
IK Chain in initial position
Alternate configurations
First configuration reached by the chain
Current root
Current sphere

Fig. 4. Generating multiple solutions



(a) Input Sketch (b) Pose 1 (c) Pose 2

Fig. 5. Returning multiple solutions

we do an inorder search starting from this node, returning one "next" solution for every request. The modified scheme is shown in Fig. 4. Example output showing two poses generated by this method is shown in Fig. 5. Note that the input sketch given in the figure can correspond to two possible actions. One is during the *back-lift* of the bat for a righthanded batsman before the stroke is made and other the *follow-through* after the stroke for a left handed batsman. Observe that Fig. 5(b) is a valid representation of the follow-through pose and Fig. 5(c) that of the back-lift pose. Details of the algorithm are skipped in this version.

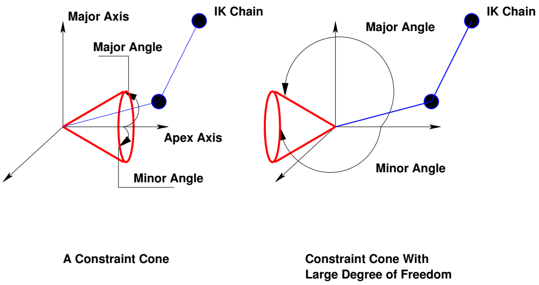


Fig. 6. The Constraint Cone

5 Experiments and Results

A sample posing result for our scheme is shown in Fig. 1. We also implemented the Jacobian based method mentioned in Section 4. By and large, based on our discussions with kinesiological experts, our method looks more natural.

For quantitative comparisons, we compared the *physical work* done against the force of gravity to bring an IK chain from initial position to final position, along with the *posing error* using each method. The posing error was computed as the perpendicular distance squared of the final end-effector position from the target ray. The work done on link $\imath$ was taken as $w_\imath = mgh_\imath$ where h is the vertical displacement of the center of mass of the link. The total work done in posing a chain was obtained by summing the contribution from each link.

About 250 experiments in different configurations were performed. A method was deemed successful if the error between the goal ray and the end-effector was less than a threshold (2% of the chain length).

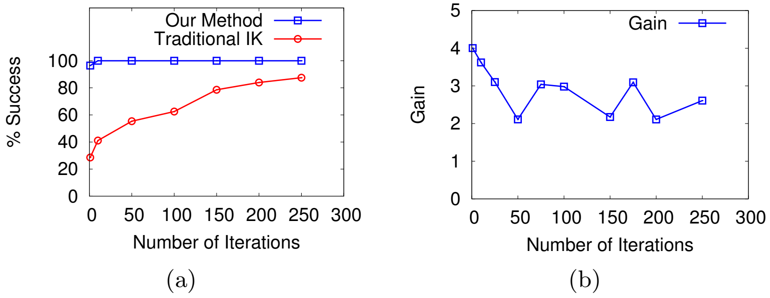


Fig. 7. (a) Error versus computation time. (b) Gain in work we achieve with respect to closest point approach.

The maximum number of iterations for which the closest point based iterative schemes were allowed to run is indicated. An equivalent amount of maximum compute time was alloted for our recursive method. As seen in Fig. 7(a) we observe that our method records success with far lesser computation time than the previous method. Also, in general our method constructs poses that require less *physical work* on the part of the character as seen in Fig. 7(b). We define gain as the ratio of the work done by our method with respect to the work done by the closest point method. Note that we are getting an improvement of atleast a factor of 2 in all cases. We compared our method with two methods: a Jacobian based blind IK method that attempts to minimize the distance with the target ray, and recent IK method[1] that targets the closest point on the target ray. Though the posing accuracy of blind IK method is comparable to our method, we got an average improvement of a factor of 1.67 in terms of the work done.

6 Conclusion and Future Work

In this work, we have implemented a scheme for constructing 3D poses from 2D sketches, photographs, and video frames. We have demonstrated that our method robustly constructs poses that look natural, and can be constructed at interactive rates. An energy efficiency paradigm has been introduced and multiple solutions are provided and in these senses too our method performs well. There are a few areas that we would like to explore further.

1. A more complete actor model that has twist and pole vector rotations. While this is a matter of detail in the forward graphics problem, it will handle issues such as head rotation in Fig 1.
2. In many applications, it may be possible to compute a *homomorphism* between the $2D$ and $3D$ skeletons, thus eliminating the need for the user to manually click corresponding end-effectors. This has several interesting issues like
   - Efficient computation of the homomorphism
   - Handling the symmetries in the structure of a character.

References

1. Chaudhuri, P., Kalra, P., Banerjee, S.: A system for view-dependent animation. In: Eurographics 2004. (2004)
2. Davis, J., Agrawala, M., Chuang, E., Popovic, Z., Salesin, D.: A sketching interface for articulated figure animation. In: Eurographics/SIGGRAPH Symposium on Computer Animation. (2003)
3. Remondino, F., Roditakis, A.: 3D reconstruction of human skeleton from single images or monocular video sequences. In: DAGM-Symposium. (2003) 100–107
4. Parameswaran, V., Chellappa, R.: View independent human body pose estimation from a single perspective image. In: CVPR (2). (2004) 16–22
5. Bowden, R., Mitchell, T., Sarhadi, M.: Reconstructing $3D$ pose and motion from a single camera view. In Carter, J.N., Nixon, M.S., eds.: In Proceedings of the British Machine Vision Conference. Volume 2., University of Southampton (1998) 904–913
6. Marler, R.T.: Development of real time multi objective optimization based posture prediction. Technical report, The University of Iowa (2004)
7. Kim, J., Abdel-Malek, K., Yang, J., Nebel, K.: Motion prediction and inverse dynamics for human upper extremities. In: SAE 2005 World Congress. (2005) 11–14
8. Welman, C.: Inverse kinematics and geometric constraints for articulated figure manipulation. Master's thesis, Simon Frase University (1993)
9. Wang, Chen, C.C.: A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. In: IEEE Transactions on Robotics and Automation. Volume 7:4. (1991) 489–499
10. Kibler, W.B., McMullen, J.: Scapular dyskinesis and its relation to shoulder pain. Journal of American Academy of Orthopaedic Surgeons (2003) 142–151

Super-Resolved Video Mosaicing for Documents Based on Extrinsic Camera Parameter Estimation

Akihiko Iketani[1], Tomokazu Sato[1,2], Sei Ikeda[2], Masayuki Kanbara[1,2],
Noboru Nakajima[1], and Naokazu Yokoya[1,2]

[1] NEC Corporation, 8916-47 Takayama, Ikoma, Nara 630-0101, Japan
iketani@cp.jp.nec.com, n-nakajima@ay.jp.nec.com
[2] Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma,
Nara 630-0192, Japan
{tomoka-s, sei-i, kanbara, yokoya}@is.aist-nara.ac.jp

Abstract. This paper describes a novel video mosaicing method based on extrinsic camera parameter estimation. With our method, a mosaic image without perspective distortion can be generated, even if none of the input image plane is parallel to the target document. Thus, users no longer have to take special care in holding the camera so that the image plane in the reference frame is parallel to the target. First, extrinsic camera parameters are estimated by tracking image features. Next, by utilizing re-appearing features, estimated extrinsic camera parameters are globally optimized to minimize the estimation error in the whole input sequence. Finally, all the images are projected onto the mosaic image plane, and a super-resolved mosaic image is generated by applying an iterative back projection algorithm. Experiments have successfully demonstrated the feasibility of the proposed method.

1 Introduction

Document and photograph digitization from a printed or drafted paper has been very important for digital archiving and personal data transmission over the internet. Flat-bed scanners are one of the most commonly used devices for this purpose. These scanners, however, are too large and heavy to be portable. Thus there has been a strong demand for a high quality digitization of documents using portable imaging devices, such as cameras on cellular phones. The problem here is the resolution of the image acquired with these devices. 2M pixel cameras enable full A4 pages to be sampled at about 150 dots per inch (dpi), whereas flat-bed scanners enable sampling at few thousand dpi.

This problem can be solved by capturing partial images of the documents as a video, and by stitching multiple frame images seamlessly into one large, high resolution image. This technique is called video mosaicing, and a number of methods have been proposed. Conventional methods estimate pairwise registration between two successive images, and construct a mosaic image by warping all the images to a reference frame (in general, the first frame). Szeliski [1] developed a method using 8-DOF projective image transformation parameters called homography. In this method, for

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 101 – 110, 2006.

every pair of consecutive frames, homography which minimizes the sum of squared differences between the two frames are estimated. This method is applicable when a target is a plane (planar image mosaicing), or optical centers of images are approximately fixed throughout the video capturing (panoramic image mosaicing). After his work, various extensions to this method have been proposed [2], [3], [4], [5], [6], [7]. One of the major extensions is the use of image features instead of all the pixels in images for reducing computational cost [2], [3], [4]. Although the computational cost is drastically reduced by these methods, a resultant mosaic image usually has a misalignment of images because of the cumulative errors in homography estimation. Some methods introduce an optimization process after homography estimation to ensure the consistency of the registration among multiple frames, and to reduce the cumulative estimation errors [6], [7].

All these methods, however, align the input images to the reference frame, thus will generate a mosaic image with perspective distortion if the image plane in the reference frame is not parallel to the target document. A mosaic image with this perspective distortion is shown in Figure 1. This is a mosaic image of an A4 sized document. The rectangular region in this figure corresponds to the image captured in the reference frame. Since the image plane in the reference frame was not set parallel to the target document, perspective distortion has occurred all over the mosaic image. In order to generate a mosaic image without perspective distortion, not only the registration among the images, but also the geometry between the document and the camera, or in other words, the extrinsic camera parameters in each frame have to be solved.

In this paper, a novel video mosaicing method based on extrinsic camera parameter estimation is proposed. The originality of this method lies in that extrinsic camera parameters, instead of homography, are estimated by applying structure from motion method [8], [9] to documents. Using estimated extrinsic camera parameters, a mosaic image without perspective distortion can be generated. Another originality of this method lies in that re-appearing image features are utilized to minimize estimation errors in extrinsic camera parameters. Note that the proposed method is based on the assumption that the target document is planar, intrinsic camera parameters are known in advance, and are fixed throughout image capturing.

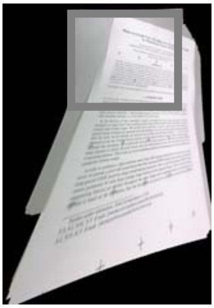
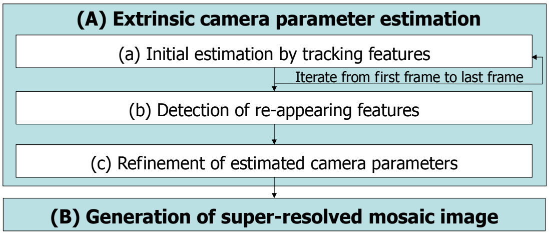


Fig. 1. Mosaic image with perspective distortion



(A) Extrinsic camera parameter estimation
(a) Initial estimation by tracking features
Iterate from first frame to last frame
(b) Detection of re-appearing features
(c) Refinement of estimated camera parameters
(B) Generation of super-resolved mosaic image

Fig. 2. Flow diagram of the proposed method

2 Video Mosaicing by Extrinsic Camera Parameter Estimation

The flow of our method is given in Figure 2. First, extrinsic camera parameters of a handheld camera is estimated (A), and a super-resolved mosaic image is then generated using the estimated parameters (B). In the following sections, first, extrinsic camera parameters and an error function are defined. The stages (A) and (B) are then described in detail.

2.1 Extrinsic Camera Parameters and Error Function

In this paper, as shown in Figure 3, the transformation matrix of the *f*-th frame is defined between the mosaic image plane and the *f*-th frame image plane as follows:

$$(a\hat{u}_{fp}, a\hat{v}_{fp}, a)^T = \mathbf{M}_f (x_p, y_p, 1)^T, \tag{1}$$

$$\mathbf{M}_f = \begin{pmatrix} c_1c_3 + s_1s_2s_3 & s_1c_2 & t_{1f} \\ -s_1c_3 + c_1s_2s_3 & c_1c_2 & t_{2f} \\ c_2s_3 & -s_2 & t_{3f} \end{pmatrix}, \tag{2}$$

$$s_i = \sin(r_{if}), \quad c_i = \cos(r_{if}) \quad (i = 1,2,3), \tag{3}$$

where

(x_p, y_p): the position of a feature *p* in the mosaic image plane,

$(\hat{u}_{fp}, \hat{v}_{fp})$: the projected position of (x_p, y_p) to the *f*-th frame image with the ideal camera model,

(u_{fp}, v_{fp}): the projected position of (x_p, y_p) to the f-th frame image in the real camera image, which is given by transferring $(\hat{u}_{fp}, \hat{v}_{fp})$ by known intrinsic camera parameters including focus, aspect, optical center and distortion parameters,

(t_{1f}, t_{2f}, t_{3f}): camera position of the f-th frame,

(r_{1f}, r_{2f}, r_{3f}): camera posture of the f-th frame,

a: a parameter.

This transformation matrix $\mathbf{M}_f$ is essentially the same as a usual extrinsic camera matrix except the omission of *z*-axis parameters, since the target object is always on the *z*=0 plane.

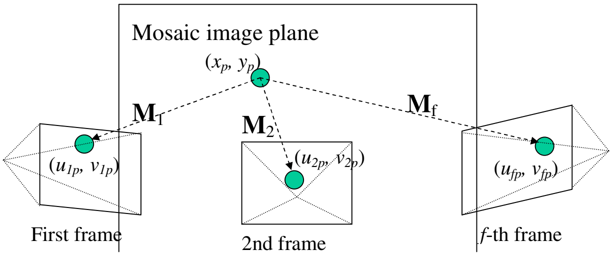


Fig. 3. Mosaic image plane and camera

In general, (u_{fp}, v_{fp}) computed by Eq. (1) and known intrinsic camera parameters does not coincide with an actually detected position (u'_{fp}, v'_{fp}) in the real image, due to errors in feature detection and extrinsic camera parameter estimation. In this paper, the squared error E_{fp} is defined as an error function for the feature *p* in the *f*-th frame as follows:

$$E_{fp} = (u_{fp} - u'_{fp})^2 + (v_{fp} - v'_{fp})^2 \quad . \tag{4}$$

The sum of E_{fp} is employed for estimating $\mathbf{M}_f$ and (x_p, y_p) in the following section.

2.2 Extrinsic Camera Parameter Estimation

As shown in Figure 2, the extrinsic camera parameter estimation method consists of three processes. The following briefly describes each process.

Initial Estimation by Tracking Features. In this process, an initial estimate of extrinsic camera parameter $\mathbf{M}_f$ is computed by tracking image features. This process is basically an extension of the method in [9].

In the first frame, $\mathbf{M}_f$ is set as an identity matrix, assuming the image plane in the first frame is parallel to the target object. For each image feature *p* in the first frame, its position (x_p, y_p) in the mosaic image plane is also computed based on this assumption. Note that even if the target object and the image plane of the first frame are not parallel to each other, they are corrected in the refinement process.

In the succeeding frames (*f*>1), $\mathbf{M}_f$ is determined by iterating the following steps until the last frame.

- **Tracking of image features:** All the image features are tracked from the previous frame to the current frame by using a standard template matching with Harris corner detector [10]. The RANSAC approach [11] is also employed for eliminating outliers.
- **Extrinsic camera parameter estimation:** The tracked position (u'_{fp}, v'_{fp}) and its position in the mosaic image plane (x_p, y_p), which is estimated in the previous iteration, are used for estimating the extrinsic camera parameter $\mathbf{M}_f$. In this step, the error function $\sum_p E_{fp}$ is minimized by Levenberg-Marquardt algorithm.
- **Estimation of feature position on mosaic plane:** The position (x_p, y_p) of each feature *p* in the mosaic image plane is refined by minimizing the error function $\sum_f E_{fp}$.
- **Addition and deletion of features:** In order to obtain accurate estimates of camera parameters, good features should be selected. The set of features is updated by evaluating the reliability of features [9].

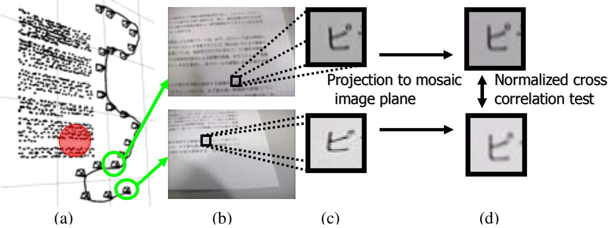

Projection to mosaic image plane
Normalized cross correlation test
(a)
(b)
(c)
(d)

Fig. 4. Detection of re-appearing features. (a) camera path, posture and feature position on mosaic image plane, (b) sampled frames of an input image sequence, (c) templates of a feature in different images, (d) templates projected to a mosaic image plane.

Detection of Re-appearing Features. When we move the camera so as to capture the whole document area without missing any part of it, the camera will make some loop-backs, revisiting certain parts of the target more than once, as shown in Figure 4 (a). Due to this camera motion, some features appear in the image more than once, as shown in Figure 4 (b). In the proposed method, these re-appearing features are detected and utilized to refine the estimated camera parameters in the following step.

Re-appearing features are detected by first projecting the templates of all the features to the mosaic image plane, and then examining the normalized cross correlation between every feature pair whose distance is less than a given threshold. This procedure is shown in Figure 4 (c) and (d).

Refinement of Estimated Camera Parameters. In this process, extrinsic camera parameters are refined in the framework of bundle adjustment [12].

Since initial estimation of extrinsic camera parameters is an iterative process, it permits the accumulation of the estimation error. Bundle adjustment is a process which jointly optimizes extrinsic camera parameters for each frame and feature position on mosaic plane, so as to minimize the cumulative estimation error.

The cumulative estimation error E to be minimized in the bundle adjustment is defined as follows:

$$E = \sum_f \sum_p E_{fp}. \tag{5}$$

This is the sum of squared distances between the re-projection of the estimated feature position onto the input image plane and its actually detected position in the input image. As the cumulative error becomes larger, so does E. E will become even larger if the image plane in the reference frame is not parallel to the target, since wrong extrinsic camera parameters are estimated for the reference frame. Thus, we minimize E with respect to the camera parameters $\mathbf{M}_f$ and the feature positions (x_p, y_p) over the whole input. With this minimization, the cumulative estimation error is reduced, and the correct extrinsic camera parameters are estimated for the reference frame, in case the image plane is not parallel to the target in the reference frame.

As for re-appearing features, all the position sequences belonging to the same re-appearing feature are merged, and treated as one single sequence in the computation of E_{fp}. This enables the extrinsic camera parameters to be optimized, ensuring the consistency of registration between distant frames, such as frames shown in Figure 4 (b).

2.3 Generation of Super-Resolved Mosaic Image

Finally, a mosaic image is generated. Here, we apply an iterative back projection algorithm [13] and generate a super-resolved mosaic image.

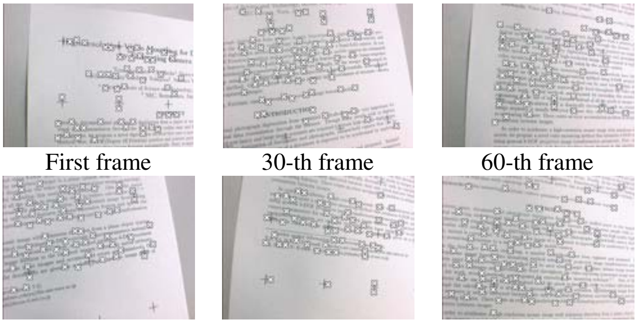
First, an initial mosaic image $S^{(0)}$ is estimated by projecting all the frame images onto the mosaic image plane using Eq. (1) with extrinsic camera parameters $\mathbf{M}_f$, and then by blending them. Starting with this initial estimate $S^{(0)}$, the imaging process is simulated to obtain a set of low-resolution images $\{I_f^{(0)}\}$, each of which corresponds to the observed input image $\{I_f\}$. If $S^{(0)}$ is the correct super-resolved image, the simulated images $\{I_f^{(0)}\}$ must be identical to $\{I_f\}$. On the other hand, as the estimation error in $S^{(0)}$ becomes larger, so does the difference between $\{I_f^{(0)}\}$ and $\{I_f\}$. Thus, the difference images $\{I_f - I_f^{(0)}\}$ are computed, and used to improve the initial estimate $S^{(0)}$ by back-projecting each value in the difference images onto its corresponding area in $S^{(0)}$. This process is repeated iteratively until the super-resolved image converges.

3 Experiment

We have developed a prototype video mosaicing system which consists of a desktop PC (Pentium-4 3.2GHz, Memory 2GB) and a calibrated IEEE1394 CCD camera (Aplux C104T). Experiments were done using this system on two kinds of plane papers. One was a printed A4 size document. The other was a photograph printed on an A4 size paper. In both papers, plus marks (+) were printed on 40mm grid positions for quantitative evaluation (described later).

3.1 Mosaicing for a Document

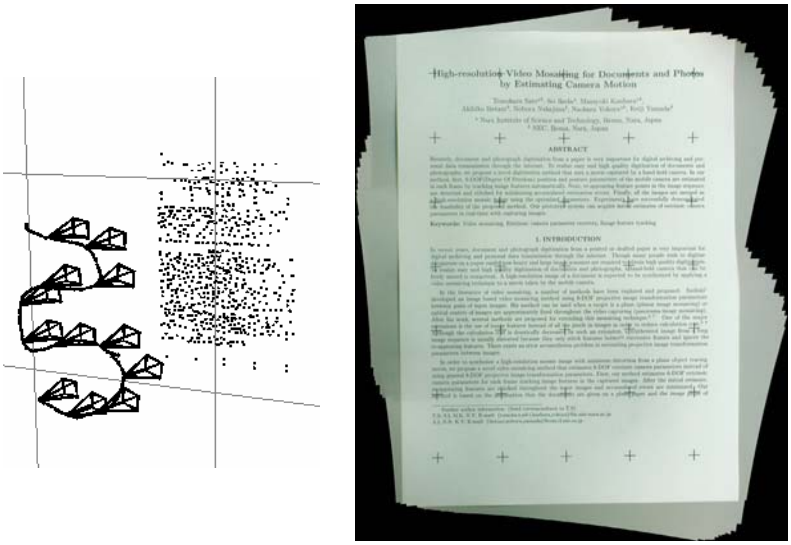
As shown in Figure 5, the target document was captured as 640×480 images of 150 frames at 15fps. Image features tracked in the initial extrinsic parameter estimation are depicted with × marks. Note that none of the input image plane was parallel to the target document. Figure 6 (a) illustrates estimated extrinsic camera parameters and feature positions on the mosaic image plane. The curved line shows the estimated camera path and pyramids show the camera postures in every 10 frames. The super-resolved mosaic image after 3 iterations is shown in Figure 6 (b). The size of the image is 2452×3002. A close shot of an input image and the super-resolved mosaic image is shown in Figure 7. As can be seen, texts which are almost unreadable in the input image are restored in the super-resolved mosaic image.



First frame 30-th frame 60-th frame

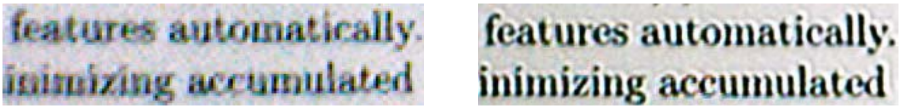
90-th frame 120-th frame 150-th frame

Fig. 5. Sampled frames of input image sequence (document)



(a) Extrinsic camera parameters (b) Generated super-resolved mosaic image

Fig. 6. Estimated extrinsic camera parameters and generated super-resolved mosaic images (document)



features automatically.
inimizing accumulated

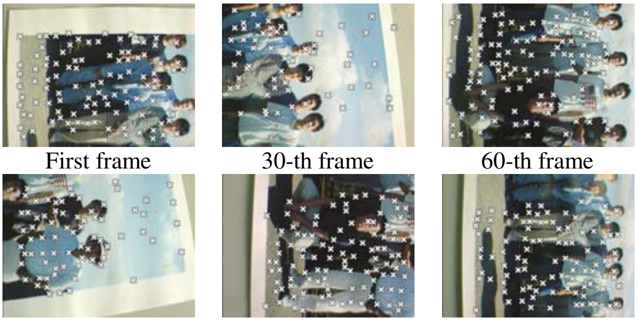
features automatically.
inimizing accumulated

(a) Input image (b) Super-resolved mosaic image

Fig. 7. Comparison between input image and super-resolved mosaic image (document)

3.2 Mosaicing for a Photograph

As shown in Figure 8, the target photograph was captured as 640×480 images of 150 frames at 15fps. Image features tracked in the initial extrinsic parameter estimation are depicted with × marks. In this experiment, the camera was held so that the input image plane in the first frame was approximately parallel to the target document. Figure 9 (a) illustrates extrinsic camera parameters and feature positions on the mosaic image plane. The curved line shows the estimated camera path and pyramids show the camera postures in every 10 frames. The super-resolved mosaic image after 3 iterations is shown in Figure 9 (b). The size of the image is 2169×2719. A close shot of an input image and the super-resolved mosaic image is shown in Figure 10. As can be seen, the frame of the glasses and the stripes on the shirt are restored in the super-resolved mosaic image.



First frame

30-th frame

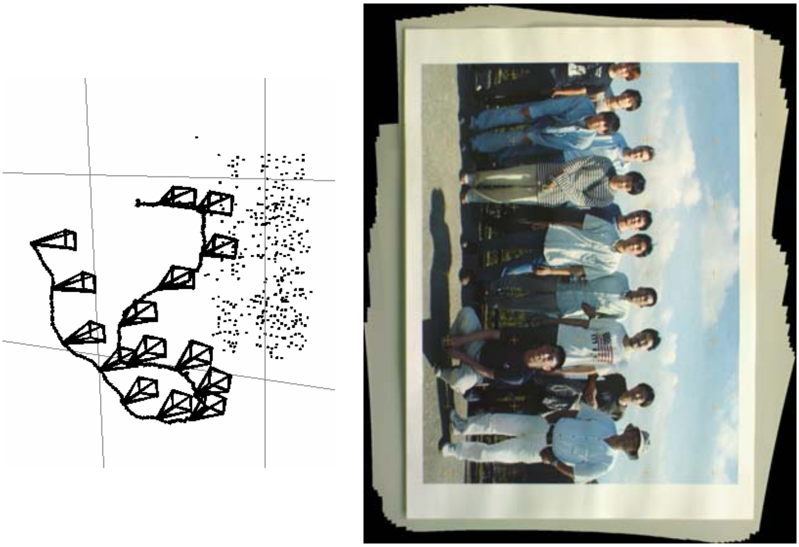
60-th frame

90-th frame

120-th frame

150-th frame

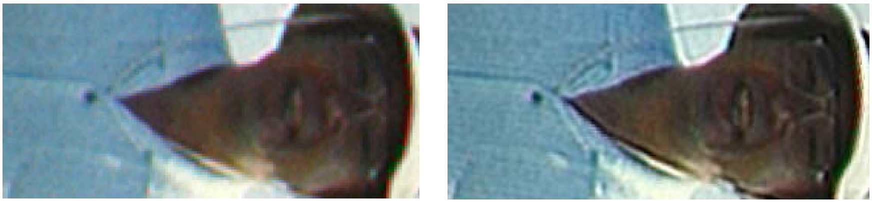
Fig. 8. Sampled frames of input image sequence (photograph)



(a) Extrinsic camera parameters

(b) Generated super-resolved mosaic image

Fig. 9. Estimated extrinsic camera parameters and generated super-resolved mosaic images (photograph)



(a) Input image (b) Super-resolved mosaic image

Fig. 10. Comparison between input image and super-resolved mosaic image (photograph)

3.3 Quantitative Evaluation of the Distortion

Finally, we quantitatively evaluated the distortions in the generated mosaic images by measuring the distances between adjacent plus marks (+) printed on the target papers. First, the positions of the plus marks were acquired manually in the generated mosaic images. The distances between adjacent plus marks were then computed in the unit of pixel. The average, maximum, minimum and standard deviation of the distances for a document and a photograph are shown in the upper and the lower row of Table 1, respectively. The percentage of each value from the average distance is also shown in parenthesis. Here, the standard deviation can be considered as the average distortion in the mosaic image. Although the average distortion of the document was a little worse than that of the photograph, both average distortions were sufficiently little for the purpose of digital archiving and personal data transmission.

The performance of our system for both sequences were almost the same as follows: 15 fps for image acquisition and initial estimation of the extrinsic camera parameters, 1 second for detecting re-appearing features, 20 seconds for camera parameter refinement, 32 seconds for initial mosaic image generation, and 122 seconds for super-resolved mosaic image generation.

Table 1. Distances of adjacent grid points in the mosaic image [pixels (percentage from average)]

Average	Maximum	Minimum	Standard Deviation
351.0(100.0)	357.9(101.9)	346.4(98.6)	2.53(0.72)
332.9(100.0)	337.6(101.4)	329.4(98.9)	1.72(0.52)

4 Conclusion

A novel video mosaicing method based on extrinsic camera parameters was proposed. With this method, a super-resolved mosaic image without perspective distortion can be generated from an image sequence where none of the input image plane is parallel to the target. Thus, users no longer have to take special care in holding the camera so that the image plane is set parallel to the target in the reference frame. Experiments with our prototype system have successfully demonstrated the feasibility of the proposed method. Our future work is to reduce the computational cost in super-resolved image generation.

References

1. Szeliski, R.: Image Mosaicing for Tele-Reality Applications. Proc. IEEE Workshop on Applications of Computer Vision (1994) 230–236
2. Chiba, N., Kano, H., Higashihara, M., Yasuda, M., Osumi, M.: Feature-based Image Mosaicing. Proc. IAPR Workshop on Machine Vision Applications (1998) 5–10
3. Takeuchi, S., Shibuichi, D., Terashima, N., Tominaga, H.: Adaptive Resolution Image Acquisition Using Image Mosaicing Technique from Video Sequence. Proc. IEEE Int. Conf. on Image Processing, Vol. 1 (2000) 220–223
4. Hsu, C.T., Cheng, T.H., Beuker, R.A., Hong, J.K.: Feature-based Video Mosaic. Proc. IEEE Int. Conf. on Image Processing, Vol. 2 (2000) 887-890
5. Lhuillier, M., Quan, L., Shum, H., Tsui, H.T.: Relief Mosaics by Joint View Triangulation. Proc. IEEE Int. Conf. on ComputerVision and Pattern Recognition, Vol. 1 (2001) 785-790
6. McLauchlan, P.F., Jaenicke, A.: Image Mosaicing Using Sequential Bundle Adjustment. Image and Vision Computing, Vol. 20 (2002) 751-759
7. Kim, D.W., Hong, K.S.: Fast Global Registration for Image Mosaicing. Proc. IEEE Int. Conf. on Image Processing, Vol. 2 (2003) 295-298
8. Tomasi, C., Kanade, T.: Shape and Motion from Image Streams under Orthography: A factorization method. Int. J. on Computer Vision, Vol. 9, No. 2 (1992) 137-154
9. Sato, T., Kanbara, M., Yokoya, N., Takemura, H.: Dense 3-D Reconstruction of an Outdoor Scene by Hundreds-baseline Stereo Using a Hand-held Video Camera. Int. Jour. of Computer Vision, Vol. 47, No. 1-3 (2002) 119–129
10. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. Proc. Alvey Vision Conf. (1988) 147-151
11. Fischler, M.A., Bolles, R.C., Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, Vol. 24, No. 6 (1981) 381-395
12. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle Adjustment - A Modern Synthesis. Vision Algorithms: Theory and Practice (2000) 298-375
13. Irani, M., Peleg, S.: Improving Resolution by Image Registration. CVGIP : Graphical Models and Image Processing, Vol. 53, No. 3 (1991) 231-239

Content Based Image and Video Retrieval Using Embedded Text

Chinmaya Misra and Shamik Sural

School of Information Technology, Indian Institute of Technology,
Kharagpur, West Bengal -721302, India
{cmisra, shamik}@sit.iitkgp.ernet.in

Abstract. Extraction of text from image and video is an important step in building efficient indexing and retrieval systems for multimedia databases. We adopt a hybrid approach for such text extraction by exploiting a number of characteristics of text blocks in color images and video frames. Our system detects both caption text as well as scene text of different font, size, color and intensity. We have developed an application for on-line extraction and recognition of texts from videos. Such texts are used for retrieval of video clips based on any given keyword. The application is available on the web for the readers to repeat our experiments and also to try text extraction and retrieval from their own videos.

1 Introduction

Text embedded in an image is usually closely related to its semantic content. Hence, text is often considered to be a strong candidate for use as a feature in high level semantic indexing and content-based retrieval. An index built using extracted and recognized text enables keyword-based searches on a multimedia database. As an example, we can identify video frames on specific topics of discussion from an educational video if the frames display corresponding text information. One of the main challenges in this work is to be able to locate text blocks in an image with complex color combinations.

Text in image and video can be classified into two broad types: (i) Caption text - also known as Graphic text or Overlay text and (ii) Scene text. Caption text as shown in Fig. 1(a), is the type of text that is synthetically added to a video or an image during editing. It serves many different purposes like display of actor list and credit in a movie, topics covered in an educational video, etc. Caption text in a video frame typically has low resolution so that it does not occlude the scene objects.

In contrast to caption text, scene text as shown in Fig. 1(b), usually occurs in the field of view of a camera during video or still photography. Examples of scene text include street signs, billboards, topics covered through presentation slides in educational videos, number plates on cars, etc. Scene text is often more difficult to detect and extract compared to caption text due to its unlimited range of font, size, shape and color. It may be noted from Figs. 1(a) and (b) that the images

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 111–120, 2006.

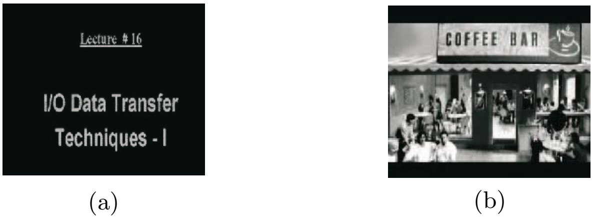


Fig. 1. (a)Caption text and (b) Scene text

containing either scene text or caption text cannot serve as a direct input to an Optical Character Recognition (OCR) system. Existing segmentation techniques built in the OCR systems are not capable of handling the complexity of color images in which such text regions are embedded. Instead, it is essential to build specialized methods for identifying the text blocks from images and video frames. Contents of such text blocks can then be submitted to an OCR for identification of the characters and words.

Our goal is to accurately extract text blocks from color images and video frames, recognize the texts using an OCR and store them as keywords in a database for indexing and retrieval.

2 Related Work

In recent years, attempts have been made to develop methods for extracting text blocks from still images and videos. Li et al [6] use a 16×16 window moved over various positions in an image. Each window position is classified as a text or a non-text block using a neural network. Text blocks identified by the classifier are then tracked across frame boundaries. This method detects text only at the block level. Jain and Yu [3] propose a method to locate text in color images using connected components. Their method can detect text only with large size and high contrast. While it is well suited for processing newspaper advertisements and web images, it is not so efficient in detecting text in complex and cluttered background. Accuracy of this approach is high for binary and gray-scale images but the system is not so accurate in locating text in full-color images.

Lienhart and Wernicke propose a multi-resolution approach to detect potential text lines in images and video frames using edge orientation [7]. This method also uses small windows to find edge orientations and a complex-valued neutral network based method to classify text regions with certain pre-defined sizes. They employ projection profiles as well as geometrical and morphological constraints for refining the text boxes. Nugroho et al [9] apply color reduction and decompose a multi-valued color image into a small number of meaningful color prototypes based on foreground color. After that, connected components are extracted from each foreground image and text and non-text components are classified with the help of stroke features. This approach works well in a limited range of characters, especially in multi-segment characters like Japanese and Chinese.

Malababic et al [8] detect artificial text in videos using a feature that captures foreground to background contrast, density of short edges of varying orientations and concentration of short vertical edges that are horizontally aligned. Various geometrical constraints are also applied for improving the result.

Sato et al [10] investigate superimposed caption recognition in news videos. They use a spatial filter to localize the text regions as well as size and position constraints to refine the detected area. This algorithm can be applied only in a specific domain, namely, news video analysis. Jung and Han [4] sequentially adds advantages of texture based methods and connected component based methods. A texture classifier detects text regions and filtering is done by the connected component based method using geometric and shape information. They detect text in images with multiple color, intensity and fonts. However, since this method processes a raw pixel values for each frame in texture classifier and performs a number of stages of filtering and refinement, it takes a lot of time for processing each image. Zhang et al [13] use a multiple hypothesis filtering approach on several binary images after decomposing a given image by color space partitioning. To find the candidate text regions they use texture and motion energy as compressed domain features. This method can be used to detect caption text from newscasts. However, it works on the assumption that most of the text is located in some predefined regions with high contrast and simple background in the video.

In contrast to the above-mentioned methods, we propose a hybrid approach in which multiple cues are used for detecting text blocks in images and videos. Further, in all of the existing methods, there is no mention of any complete system being developed using the text extraction techniques. We feel that along with the development of new algorithms, it is equally important to be able to demonstrate the results. For this purpose, we have built a video retrieval system based on embedded text, which is available on the web. Interested readers will be able to repeat our experiments and also perform their own retrievals using this application.

The rest of the paper is organized as follows. In the next section, we give a description of our system. The results are presented in section 4 and we conclude in the last section of the paper.

3 Hybrid Approach to Text Extraction

In this section, we first give an overview of our system followed by a detailed description of the building blocks.

3.1 Overview of the Approach

The input to our system can either be a still image or a video decomposed into frames. We first use a color reduction step in which the input is converted into a 64-color image. This step is necessary since there can be a large number of colors present in an image. Individual color level processing makes the system both inefficient as well as sensitive to noise. We next determine the Regions of

Interest (ROIs) - Regions in the image where text could potentially be located. This step, while meant to speed-up subsequent searches, should not filter out the text regions. Care is, therefore, taken to ensure that only those regions that are certainly of type non-text are eliminated.

After identification of the regions of interest, geometrical and morphological features are extracted from each ROI. A multilayer perceptron (MLP) is used as a feature-based classifier to determine if the ROI contains text or non-text blocks. It should be noted that at this stage, we identify an entire ROI to either belong to a text region or to a non-text region and not its individual components. After classification of an ROI as text or non-text, the potential text regions are subjected to a connected component analysis for reducing the false positives. Connected components of the regions of interest so far marked as text, are examined for the existence of specific text features. If such features are not present in the connected components, they are eliminated. The remaining components are marked as text blocks. These text blocks are next given as input to an OCR. The OCR output in the form of ASCII characters forming words is stored in a database as keywords with frame reference for future retrieval.

3.2 Detailed Description of the Steps

I Frame Extraction. A text can be detected from static images or videos. For video sequences, since text must be present for at least half a second for viewers to read the contents, we use only I-frames for text extraction from videos with the typical IBBPBBPBBPBB sequence at a rate of 30 frames per second. Any text which occurs in a video for duration less than the time gap between successive I-frames, is not useful to the viewers as well and hence need not be considered. If a video follows any other frame sequence, we extract every twelfth frame for text extraction. This step is not required for processing still images.

Color Reduction. Color reduction is an important pre-processing step for text extraction from complex still images and videos. We perform color reduction by taking the 2 higher order bits from the R, G and B color bands. Now each image contains only 2^6 color combinations instead of 2^{24}. In Fig. 2(a) and Fig. 2(b) we show an original image and the corresponding color reduced image, respectively.

After color reduction, each pixel has a color value v $\epsilon\psi$ where ψ ={0,1,2, 3, ...(V-1)}, V being the total number of colors. If only two higher order bits

(a) (b)

Fig. 2. (a)Original image and (b) Color reduced Image are used, V=64. We use the color-reduced image for the identification of regions of interest.

Region of Interest Identification. We next identify potential regions in the image where text may be located. Identification of such regions of interest helps in speeding up the process of text extraction. In the text regions, there are high densities of foreground pixels for some meaningful color plane. A projection profile of an image region is a compact representation of the spatial pixel content distribution. Horizontal projection profile (HPP) for a given color is defined as a vector of the pixel frequency over each row for that color. Vertical projection profile (VPP) is defined similarly. A threshold for HPP, T_H=8, and a threshold for VPP, T_V=2, is set to refine the region of interest (ROI). A text is expected to be located in image regions where the count of pixels for a given color in the horizontal direction is greater than T_H and the count of pixels for the same color in the vertical direction is greater than T_V. Texts usually do not have fixed sizes in images and video frames. However, more than 99% of all texts are less than half the image height and at least greater than 4 pixel in height to make them legible.

Geometrical and Morphological Feature Extraction. For each ROI, a number of features are extracted for each color. Before feature extraction, the regions of interest are binarized as follows.

Let v_{ij} denote the color value of the pixel (i,j) after color reduction. For a given color v_k, v ϵ ψ, binarization is done as follows:

```
for i=1 to ROI_Height
        for j = 1 to ROI_Width
            if v$_{i,j} = v_k$
                $Set\ \ v_{i,j} = 1$
            $else$
                $Set\ \ v_{i,j} = 0$
```

Thus, when we process any given color, we set all pixels in the ROI of that color to 1 and the rest to 0.

A total of 7 features are extracted which are briefly mentioned below.

i. Foreground Pixel Density - It is the number of pixels per unit area whose binarized value is 1.
ii. Ratio of Foreground Pixel Density to Background Pixel Density - Background pixel density is calculated in a manner similar to foreground pixel density described above.
iii. Edge Pixel Density - Edge pixels are defined as the ones for which one of its eight neighbors has a binarized value of 0.
iv. Foreground Pixel to Edge Pixel Ratio - Ratio of foreground pixel density to edge pixel density
v. Horizontal Edge Pixel Density
vi. Vertical Edge Pixel Density
vii. Diagonal Edge Pixel Density.

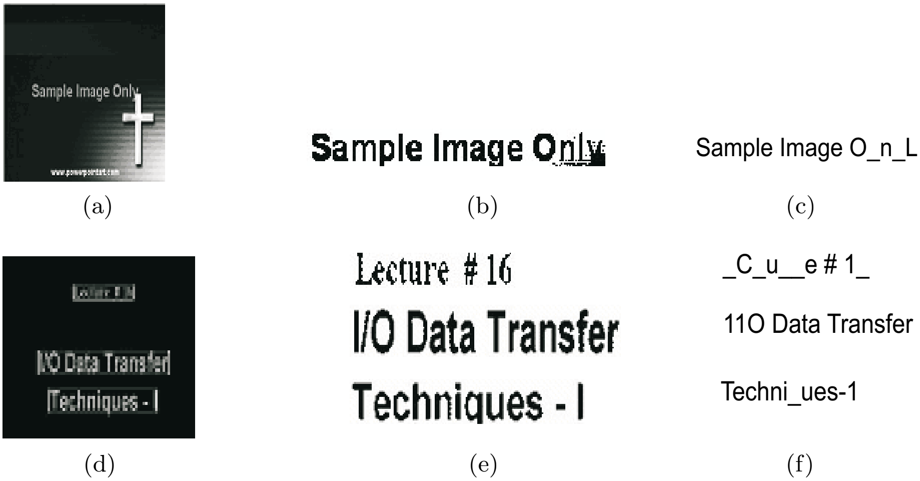
MLP Based Classification. The geometrical and morphological features extracted from each region of interest are next used for classification by a multilayer perceptron. In the learning phase, we use features extracted from a set of images containing both text and non-text regions. Such regions are manually checked and assigned the corresponding ground truth. 200 text regions and an equal number of non-text regions are used for training the MLP. The MLP contains 7 inputs, one hidden layer of 10 units and 1 output. The output represents whether the input block contains text or non-text. The MLP was trained with different initial conditions and was found to have similar performance in each case.

Connected Component Analysis. In order to reduce the number of false positives after MLP based classification, we introduce connected component analysis as a post-processing step. The following heuristics are applied to filter out possible non-text blocks from the list of connected components.

i. Text lines are usually separated from image boundaries.
ii. Base and ceiling of the text components are in the same line.
iii. At least four text blocks are present in an ROI for meaningful text representations.

At the end of this post-processing step, most of the non-text blocks are removed and the rest of the regions of interest are expected to contain only text.

OCR Based Identification. Text blocks in each region of interest are given as input to an OCR for recognition. The generated outputs from the OCR are ASCII characters, which are stored in a database as keywords for future indexing and retrieval. In Fig. 3, we explain the process of text recognition in detail. Fig. 3(a) shows an ROI identified as a text block. This ROI is separated out from



```
Sample Image O_n_L
```

```
_C_u__e # 1_
11O Data Transfer
Techni_ues-1
```

(a) (b) (c) (d) (e) (f)

Fig. 3. (a) Image with ROIs identified (b) Binarized text block (c) OCR output (d) Image with multiple ROIs (e) Multiple binarized text blocks (f) OCR output for multiple text blocks

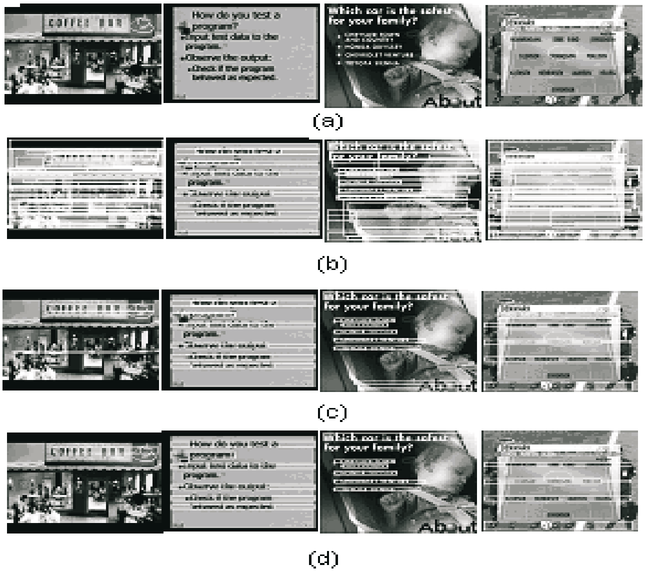


Fig. 4. Various stages of text extraction from an image (a) Original (b) After ROI detection (c) Output of MLP based classification (d) Final result

the rest of the image and binarized as shown in Fig. 3(b). When this ROI is given as input to the OCR, the corresponding ASCII output is shown in Fig. 3(c). It is observed that while the text extraction part of our system detects the text blocks accurately even in a complex background, the OCR sometimes fails to recognize the text correctly. As seen in Fig. 3(c), the last word was mis-recognized due to the presence of noise. Another example image with multiple ROIs containing caption text is shown in Fig. 3(d). Here also the text regions have been identified correctly as shown in Fig. 3(e). The corresponding OCR output is shown in Fig. 3(f). While a specific off-the-shelf OCR is currently being used in our work, it is expected that the character recognition accuracy and hence the overall system performance will improve further if a better OCR is used. The effect of the hybrid approach on the quality of text extraction is explained using Fig. 4. In Fig. 4(a), we show four original images of varying complexity. The detected regions of interest are shown in Fig. 4(b). It is observed that at this stage, recall is very high (greater than 90%) but there are a number of false positives. The MLP based classifier can correctly detect most of the text blocks and eliminate a large number of non-text blocks. The output of the MLP is shown in Fig. 4(c). At this stage, the precision has improved considerably. In Fig. 4(d) we show the image after the connected component based post-processing step. It is seen that the final result has high recall as well as precision.

3.3 Web-Based Video Retrieval System

We have developed a web-based on-line video retrieval system using embedded text. It should be noted that to facilitate blind review, the web site address has not been mentioned in this initial version of the paper. However, it will be made available in the accepted version. To test the accuracy of any system, users are often interested in retrieving frames from their own video files. To facilitate this, we provide a utility to upload an external video file in our system. From the video, keywords are extracted and stored in a database in a fully automated manner. The user can then query the database with his choice of keywords. Sets of consecutive video frames containing the keywords are retrieved from the database. A short video clip is generated from each set of consecutive frames and returned to the user for viewing. Thus, user gets back a collection of short video clips containing his choice of keywords. To the best of our knowledge, this feature is unique in our work and is not available in any other text extraction system available in the research domain.

4 Results

In this section, we present quantitative results on the performance of the text extraction system. The performance can be measured in terms of true positives (TP) - text regions identified correctly as text regions, false positives (FP) - non-text regions identified as text regions and false negatives (FN) - text regions missed by the system. Using these basic definitions, recall and precision of retrieval can be defined as follows:

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN}) \text{ and Precision} = \text{TP}/(\text{TP}+\text{FP})$$

While the above definitions are generic, different researchers use different units of text for calculating recall and precision. Wong and Chen [12] consider the number of characters while some of the other authors count the number of text boxes or text regions [1,6]. Jain and Yu [3] calculate recall and precision by considering either characters or blocks depending on the type of image. We adopt the second definition in which we consider the text regions as units for counting. The ground-truth is obtained by manually marking the correct text regions.

We have calculated recall and precision on a large number of text-rich images. For video processing, we have tested the system on different types of mpeg videos such as news clips, lecture clips and commercials. The videos contain both caption texts as well as scene texts of different font, color and intensity.

Table 1 shows the performance of our proposed method on four types of video. It is seen that our method has an overall average recall of 82% and precision of 87%. Another important consideration is the quality and complexity of pictures for evaluation. Jain and Yu consider large fonts in web images, advertisements and video clips [3]. Kim [5] does not detect low contrast text and small fonts. Li et al [6] use text with different complex motions. Zhang et al [13] as well as Sato et al [10] detect only caption text in news video clips. We are able to detect text under a large number of different conditions like text with small fonts, low intensity, different color and cluttered background, text from noisy video, News caption with horizontal scrolling and both caption text and scene text.

Table 1. Recall and precision of text block extraction

	News	Sports	Lectures	Commercials
No. of text blocks	780	144	120	3241
TP	624	120	96	288
FP	52	60	24	36
FN	156	24	24	36
Recall (%)	80	83.3	80	88
Precision (%)	92	66.6	80	88

Table 2. Execution time of text extraction

	Proposed	[12]	[4]	[11]
Machine used	PIV	Sun Ultra sparc	—	PIV
Image size	—	320*240	320*240	—
Processing Time(sec)	0.14	1.2	0.47	1.7

The primary advantage of the proposed method is that it is very fast since most of the computationally intensive algorithms are applied only on the regions of interests. Table 2 shows processing time for different types of video clips using a 2.4 GHZ Pentium-IV machine. We show comparative time required by different algorithms including those proposed in [4], [11] and [12]. For our algorithm the average is taken over a number of different image sizes. It is seen that our algorithm requires the least time for processing each frame. Since we process every I-frame which occurs at the rate of about 3 per second, we are able to achieve real time processing speed in our system

5 Conclusions

We have presented a hybrid approach for the detection of text regions and recognition of texts from images and video frames. It can detect both scene text and caption text. A content-based video retrieval system has been developed in which keywords are extracted from video frames based on their textual content. The keywords are stored and indexed in a database for retrieval.

We plan to extend our work in the compressed domain processing to make it even faster. A more accurate OCR will also improve the quality of retrieval further.

Acknowledgement

The work done by Shamik Sural is supported by research grants from the Department of Science and Technology, India, under Grant No. SR/FTP/ETA-20/2003 and by a grant from IIT Kharagpur under ISIRD scheme No. IIT/SRIC/ISIRD/2002-2003.

References

1. L. Agnihotri and N. Dimitrova: Text Detection in Video Segments. Proc. of Workshop on Content Based Access to Image and Video Libraries, pp. 109-113, June 1999
2. Y. M. Y. Hasan and L. J. Karam: Morphological Text Extraction from Images. IEEE Transactions on Image Processing. Vol. 9, Nov. 2000
3. A. K. Jain and B. Yu: Automatic Text Location in Images and Video Frames. Pattern Recognition, Vol. 31, No.12, pp. 2055-2076, 1998
4. K. Jung and J. H. Han: Hybrid Approach to Efficient Text Extraction in Complex Color Images. Pattern Recognition Letters Vol. 25, pp. 679-699, 2004
5. H-K. Kim: Efficient Automatic Text Location Method and Content-Based Indexing and Structuring of Video Database. Journal of Visual Communication and Image Representation, Vol. 7, No 4. pp. 336-344, Dec 1996
6. H. Li, D. Doerman and O. Kia: Automatic Text Detection and Tracking in Digital Video. IEEE Transactions on Image Processing. Vol. 9, pp. 147-156, Jan. 2000
7. R. Lienhart and A Wernicke: Localizing and Segmenting Text in Images and Videos. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 4, pp. 256-268, April 2002
8. J. Malobabic, N. O'Connor, N. Murphy and S. Marlow: Automatic Detection and Extraction of Artificial Text in Video. Adaptive information cluster, center for digital video processing, Dublin city university, Dublin city University, 2002
9. A. S. Nurgroho, S. Kuroyanagi and A. Iwata: An Algorithm for Locating Characters in Color Image using Stroke Analysis Neural Network. Proc. of the 9th International Conference on Neural Information Processing (ICONIP'02), November 18-22, 2002
10. T. Sato, T. Kanade, E. Hughes and M. Smith: Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions. Multimedia Systems, Vol. 7, pp. 385-394, 1999
11. J.C. Shim, C. Dorai and R. Bolle: Automatic Text Extraction from Video for Content-Based Annotation and Retrieval. Proc. of the 14th International Conference on Pattern Recognition, Vol. 1, pp. 618-620, Brisbane, Australia, August 1998
12. E. K Wong and M. Chen: A New Robust Algorithm for Video Extraction. Pattern Recognition, Vol. 36, No. 6, pp. 1397-1406, June 2003
13. D. Zhang, B. L. Tseng, C. Y. Lin and S. F. Chang: Accurate Overlay Text Extraction For Digital Video Analysis. Columbia University Advent Group Technical Report, 2003

Object Tracking Using Background Subtraction and Motion Estimation in MPEG Videos

Ashwani Aggarwal, Susmit Biswas, Sandeep Singh,
Shamik Sural, and A.K. Majumdar

Indian Institute of Technology, Kharagpur,
West Bengal -721302, India
shamik@sit.iitkgp.ernet.in, akmj@cse.iitkgp.ernet.in,
Ashwani.Aggarwal@iitkgp.ac.in

Abstract. We present a fast and robust method for moving object tracking directly in the compressed domain using features available in MPEG videos. DCT domain background subtraction in Y plane is used to locate candidate objects in subsequent I-frames after a user has marked an object of interest in the given frame. DCT domain histogram matching using Cb and Cr planes and motion vectors are used to select the target object from the set of candidate objects. The target object position is finally interpolated in the predicted frames to obtain a smooth tracking across GOPs.

1 Introduction

Visual content in a video can be modeled as a hierarchy of abstractions. At the lowest level are the raw pixels with color information leading to lines, curves, edges, corners and regions. At the highest level are the human level concepts involving one or more objects and relationships among them. The first step in high level video processing is to identify the objects present in a scene. The next step is to see how these detected objects move with respect to each other. The above two problems combined, can be termed as "Object Tracking".

An important application of object tracking is video surveillance [1]. Airports, train stations, departmental stores, religious places, courts and public buildings are only a few examples of places where video surveillance has an extremely high priority. In addition to this, military, astronomy, navigation, road/air traffic regulation, medical imaging, augmented reality and robotics are some of the other major applications of object tracking [2],[3],[4].

There are primarily two sources of information in video that can be used to track objects: visual features (such as color, texture and shape) and motion information. A typical strategy is to segment a frame into regions based on color and texture information first, and then merge regions based on similar motion subject to certain constraints such as adjacency [5]. Extraction of these two types of information can be done either in the pixel domain or in the compressed domain. Tracking in videos especially from large databases, requires an enormous

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 121–130, 2006.

amount of processing since the videos are usually stored in a compressed form and must be decompress every time any processing is done on the video frames for object tracking. In order to reduce computational time and save processing resources, it is imperative that the tracking takes place in compressed domain itself.

Sukmarg and Rao [6] performed object detection and segmentation using color clustering, region merging based on spatiotemporal similarities, and background/foreground classification. The features extracted from the blocks of segmented object in compressed domain are used for fast object tracking.

Mezaris et al [7] exploit the motion vectors available directly from the MPEG stream for object tracking in the compressed domain. Park and Lee [8] use tracking using Mean Shift algorithm along with motion vectors. Yoo and Lee [9] suggested it is not possible to get sequential motion flow directly between compressed B-frames. Hence some kind of interpolation is normally used. Kartik et al [10] perform a block matching technique over the frames. This is combined with an adaptive block based approach for estimating motion between two frames.

In this paper, we propose a novel object tracking technique from MPEG videos. We employ a background subtraction method in the compressed domain using DC values of the Discrete Cosine Transform (DCT) coefficients of luminance blocks in the I-frames. To distinguish a target object from a set of candidate foreground objects, we use histogram comparison on color components in I-frames (Cb and Cr blocks) and distance between centroid of the candidate object and projected object using motion vectors. The object positions in the intermediate P and B-frames are obtained by interpolation.

In the next section we describe the motion estimation, background subtraction and interpolation algorithm in detail. We present the experimental results in section 3. The conclusions are drawn in the last section of the paper.

2 Object Identification and Interpolation

The proposed scheme for object tracking is mainly concerned with video surveillance applications where the camera is assumed to be fixed with a fairly wide angle of view. The algorithms presented in this paper consider supervised tracking of objects in which a user marks an object in an I-frame. The marked object is tracked by our algorithm in subsequent frames till the object disappears from the field of view of the camera. Such applications typically have a model of the background which is effectively used in our approach for identification and tracking of objects. It should be noted that, although the background is considered to be fixed, our system is robust in the presence of variations in the lighting conditions and extraneous movements.

Our method for object tracking can be divided into four broad steps, namely, background subtraction, candidate object identification, target object selection and motion interpolation. All the four steps are executed directly on MPEG compressed videos. We consider a typical 12-frame Group-of-Pictures (GOP) sequence: IBBPBBPBBPBB in our discussions.

2.1 Motion Estimation

In an MPEG video, the I-frames are intra coded while the P and B frames are motion compensated and DCT is carried out only for error components after motion estimation. The macroblocks for which motion estimation cannot be carried out are also stored as intra coded macroblocks in these predicted frames.

Since we consider supervised tracking, that is, a user marks an object in an I-frame which is subsequently tracked by the system, let us assume that the user chosen pixel domain area of the object is covered by $A_p[(x_{min}, y_{min}), (x_{max}, y_{max})]$ and the equivalent compressed domain area of the object covered be given by $A_c[(p_{min}, q_{min}), (p_{max}, q_{max})]$.

We parse the next predicted frame (whichever P or B) as per the frame sequence and extract motion vectors of all the macro blocks, which are in A_c. If there are 'n' macroblocks in A_c, we get a set of n vectors (right, down) corresponding to the macroblocks. These n motion vectors do not actually provide the real optical flow since they are often inaccurate and hence, some filtering is needed. In our work we use a 'Mode filter' in order to straighten up noisy vectors and thus eliminate this problem.

We calculate the 'Mode Motion Vector' through the extracted set of 'n' forward motion vectors. This is in reference to the previous I/P frame. It gives the displacement of the window consisting of the tracked object in the current predicted frame (B/P) frames and the window coordinates are updated accordingly.

The window enclosing the object keeps updating its coordinates as the predicted frames keep coming and the tracked object moves frame by frame. This continues till the time a new reference frame within the same GOP is encountered. When a new reference frame within the same GOP is encountered, the target window in the new reference frame becomes the updated reference window. All subsequent predicted frames track the object based on the motion vectors held by them as dictated by the latest reference frame.

The above process continues till the 1^{st} I frame of the next GOP arrives. Normally in the bit stream order of a compressed video, the I frame of a GOP is followed by the B frame of the previous GOP, which are backward predicted, from this I frame. Also since the I frame does not have any motion vectors, we use the 'Backward Motion vector' of the last B frame in the display order to track the object into the I frame of the following GOP.

We consider the window of object position of the last B frame. The backward motion vectors of the macroblocks in this area give an approximate position of the window. Let the macroblock in the previous I frame (i_{th} frame in the sequence) be denoted by the term $P^i_{j,k}$, where (j,k) is the position of the macroblock in the frame. So,

$$P^m_{i,j} = P^n_{x,y} - MV((x-i),(y-j)) \tag{1}$$

Here m_{th} frame is a B frame in reference to the n_{th} I frame being described by the motion vector ((x -i), (y -j)). We can get the position of this macroblock by

$$P^n_{x,y} = P^m_{i,j} + MV((x-i),(y-j)) \tag{2}$$

To cross GOP boundary, the image region on the new I frame is obtained again with the difference that the backward motion vector mode of the last B frame is used, and the B frame pixel domain rectangle itself is treated as the reference rectangle.

2.2 Background Subtraction

We use only the successive I-frames for tracking in our algorithm and thereafter interpolate the object motion in the intermediate P and B frames. We initially acquire a DCT image of an I-frame representing the background, which is used as the reference image. Thereafter, all subsequent DCT images are compared with this reference image to segment the foreground object. The background image is based on the model of the application and is updated from time to time whenever there is a permanent change in the background.

Out of the three-color components Y, Cb and Cr, we read the DCT values of only the Y plane from an MPEG file and consider DC values of all the macroblocks contained in the background frame. These DC values contain the average luminescence for the entire frame at the macroblock level. Thus, we effectively create a DC image of the frames under comparison with only the Y component taken into consideration.

Let $I_k(M, N)$ be the k^{th} I-frame in a video sequence having a height of M pixel and width of N pixel and let the DC image of luminance blocks of this frame be denoted as $I_k^{DC,Y}(M/8, N/8)$. Width and height of this Y DC image is reduced by a factor of 8 as compared to the original I frame, since only 1 DC value is used to represent 8X8 pixel macroblock. The value of the $(i, j)^{th}$ element of the luminance DC image is given by

$$I_k^{DC,Y}(i,j) = \frac{C_u C_v}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} I_k^Y(8i + x, 8j + y) \tag{3}$$

The luminance DC image of the background frame I_0 can be similarly determined. If an object has been marked in the k^{th} I-frame, we identify its position in the $(k+1)^{th}$ I-frame by subtracting $I_{k+1}^{DC,Y}$ from $I_0^{DC,Y}$. Thus, we obtain a difference image in the form of block-wise absolute difference of the Y DC background frame and the Y DC $(k + 1)^{th}$ frame. The difference image can be represented as $\triangle I_{k+1}^{DC,Y}(M/8, N/8)$ where the value of the $(i, j)^{th}$ element is given as follows

$$\triangle I_{k+1}^{DC,Y}(i,j) = \begin{cases} I_{k+1}^{DC,Y}(i,j) & ,if I_{k+1}^{DC,Y}(i,j) - I_0^{DC,Y}(i,j) \geq T \\ 0 & ,otherwise \end{cases} \tag{4}$$

Here T is the threshold of difference.

It should be noted here that the difference image $\triangle I_{k+1}^{DC,Y}$ is expected to show high values corresponding to the image regions where the object has moved in the $(k + 1)^{th}$ I-frame. However, the difference image may also show high values in the regions where either a different moving object (not marked by the user) is present or there is a change in the background due to variation in lighting condition or presence of spurious movements like that of tree leaves and clouds. To make our system robust against such noise, we use an *adaptive threshold* (T) to compare the two frames, namely, I_{k+1} and I_0. We used T as 10% of the average of the sum of the DC component of all the blocks in the frame.

In case the difference between the DC values of two macro blocks having the same coordinates in frames under comparison have value greater than the threshold, then the macroblock of the target image is considered to be a part of the foreground and could be part of the tracked object. If the difference value is less than the threshold, we conclude that there has been no change in background for that particular block. After performing the subtraction of the luminance DC images, we generate an image where only the regions showing possible presence of foreground objects are retained. For all other regions, we set the values of all 64 DCT values (one DC and sixty three AC values) to zero. Thus, the process of background subtraction is equivalent to the application of a compressed domain mask on the entire image where the background subtraction mask ($Mask_{BS}$) is given by

$$Mask_{BS}(p,q) = \begin{cases} 1\ , if|I_{k+1}^{DC,Y}(p,q) - I_0^{DC,Y}(p,q)| \geq T \\ 0\ , otherwise \end{cases} \qquad (5)$$

On application of the mask, at positions where the mask has a value of 1, the original DC values are retained. The complete algorithm for compressed domain background subtraction can now be written as shown in Fig. 1.

```
For all I Frames
Begin
    For j ← 1 to Y_DC_Image_Height
        For k ← 1 to Y_DC_Image_Width
        Begin
            Compute $Mask_{BS}(j,k)$
            $I_{k+1}^{DC,Y}(j,k) \leftarrow I_{k+1}^{DC,Y}(j,k) \bullet Mask_{BS}(j,k)$;
        end
end
```

Fig. 1. Algorithm for background subtraction

2.3 Candidate Objects Identification

We have explained in the last sub-section how background subtraction in the $(k+1)^{th}$ I-frame is done in the compressed domain. It is also mentioned that, since we perform threshold of the DC component of luminance values, we may get multiple regions in the $(k+1)^{th}$ I-frame which show high difference values. In the next step, we locate these candidate objects in the difference image. For this, the difference image obtained by applying the $Mask_{BS}$ is used to construct a binary image. We determine the bounding boxes of the candidate objects using a variant of DFS traversal for finding all the connected components (CC) from the binary image. Regions with bounding boxes less than a threshold size are filtered out.

It should be noted that the CC analysis is done in-memory since the output of the background subtraction mask is available in the data structures and no further reading/uncompress is required.

At the end of this step, we have a set of candidate objects from which the target object is selected using histogram matching as explained in the next subsection.

2.4 Target Object Selection

At this stage, our goal is to identify the tracked object from all the candidate objects in the foreground. We make use of the color and motion vector information available in the compressed domain for target object selection. We extract DCT coefficients of Cb and Cr components of all the macroblocks enclosed in the bounding boxes of the candidate objects. We also project the intended object (marked object) to the $(k+1)^{th}$ I-Frame by using motion vectors and then calculate the difference between the centroid of this projected object and the candidate object.

The objective here is to select that particular foreground object in the $(k{+}1)^{th}$ I-frame as the target object which has the maximum color-based similarity and minimum distance with the object marked by the user in the $(k)^{th}$ I-frame. To achieve this, we create '*Reference histogram*' of DCT coefficients of Cb and Cr components of the object marked by the user.

The histogram is generated from the DC value and first eight AC values of the Cb and Cr components since higher frequency AC values beyond the first eight do not contain much information and are often found to have very low/near zero values. From each of the 9 components (1 DC and 8 AC) of the two color planes, we create 128 bin histograms. Thus, we get 18 sets of reference histograms for the marked object.

For all the candidate objects identified in the previous step described in subsection 2.3, similar histograms are created called the '*Target Histogram*'. The reference histograms are then matched with all the target histograms. The difference between the Reference Histogram and Target Histogram is stored for a candidate object "i" as DiffHis(i).

The projection of the object marked by the user in the $(k)^{th}$ I-frame is taken on the $(k+1)^{th}$ I-frame. Its centroid is calculated by

$$x_m = (x_1 + x_2)/2, \quad y_m = (y_1 + y_2)/2 \tag{6}$$

Here (x_m, y_m) is centroid of the projected object. In the same way centroid of a candidate object 'i' is calculated and its difference with the centroid of the projected object is stored as DiffCen(i). Total distance is calculated for all the candidate objects by

$$Dis(i) = w_1 \bullet DiffHis(i) + w_2 \bullet DiffCen(i) \tag{7}$$

Here *Dis(i)* is the total difference of the candidate object "i",' w_1 is the weight given to the Histogram Difference and w_2 is the weight given to Centroid difference.

The candidate object with the minimum '*Dis()*' is selected as the one corresponding to the target object. The location of this object is considered to be the location where the object marked in the $(k)^{th}$ I-frame has actually moved in the $(k+1)^{th}$ I-frame. The same process is repeated for identifying the locations of the object in all the subsequent I-frames. The algorithm for target object selection is shown in Fig. 2. In the algorithm we consider *CreateHist()* to be a function that creates eighteen 128-bin histograms for each candidate object from Cb and Cr values. *HistDiff()* returns the difference between two histograms and *CenDiff()* returns the difference between the centroid of two objects.

```
Input : CbImg, CrImg, NumCandObj, CandObj[], ProjObj, RefHist[0..17][0..127]
Output : TargObj
Algorithm :
MinDiff ← INFINITY
for i ← 1 to NumCandObj
begin
    TargHist[i] ← CreateHist ( CbImg, CrImg, CandObj[i] );
    Diff His[i] ← HistDiff ( TargHist[i] , RefHist );
    DiffCen[i] ← CenDiff ( CandObj[i], ProjObj );
    Dis[i] ← $w_1$ • DiffHis[i]  +  $w_2$ • DiffCen[i]
    If ( Dis[i] $\leq$ MinDiffSum )
    begin
        TargObj ← CandObj[i];
        MinDiffSum ← Dis[i];
    end
end
return TargObj
```

Fig. 2. Algorithm for target object selection

A weighted sum of differences of the target and reference histograms with higher weights given to DC values and less to the AC values. The weights are chosen in such a way that the DC value, which most prominently conveys color information, is given the maximum weightage. It is followed by lower frequency AC values, which convey coarse texture and shape information and then higher frequency AC values [11], in decreasing order of importance.

2.5 Object Interpolation

We exploit the compressed domain features, namely, DCT coefficients of the luminance blocks and chrominance blocks for background subtraction and target object identification in subsequent I-frames after a user has marked it in a given I-frame. We interpolate the positions of the tracked object in the intermediate frames. Consider a GOP sequence IBBPBBPBBPBBI. Let the number of predicted frames be denoted by N. We divide the displacement vector between two I -frames by N. Let $\partial\bar{v}$ be the unit displacement vector per frame is given by

$$\partial \bar{v} = \frac{ObjectPos(I_{k+1}^{DC,Y} - I_{k}^{DC,Y})}{N} \tag{8}$$

The marking window in the intermediate frames, enclosing the object, is updated in accordance with their sequence number and temporal reference between the two I-frames. The interpolated rectangle coordinates is the predicted object's location. For intermediate P and B frames denoted by F_i, the object position is obtained using Eq. 9

$$ObjPos(F_i) = ObjPos(I_k) + \partial \bar{v} \bullet (TR(F_i) - TR(I_k)) \tag{9}$$

Here $TR(F_i)$ represents the temporal reference of the frame F_i which comes between I_k and I_{k+1}.

3 Experimental Results

We have performed large-scale experiments with our object tracking system. The video sequences for experiments were shot in outdoor as well as indoor environments to cater to different situations like change in background, object size, illumination, etc. We converted these video clippings into a standard MPEG video. Various kinds of moving objects were used for testing, including cars, slow and fast moving humans as well as multiple objects.

We show results of object tracking in a number of complex situations. In Fig. 3, there are two objects of same color, but only one is marked by the user, which was successfully tracked.

In Fig. 4, we show tracking results under indoor lighting conditions.

Quantitative result on accuracy is shown in Table 1. PA denotes the combined background subtraction and motion estimation based method as proposed in this

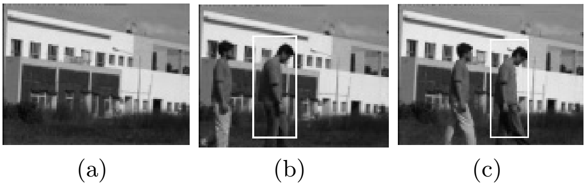

(a) (b) (c)

Fig. 3. Tracking of object in outdoor location (a) Background (b) Object marked by user (c) Tracked object

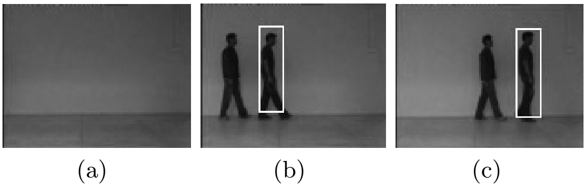
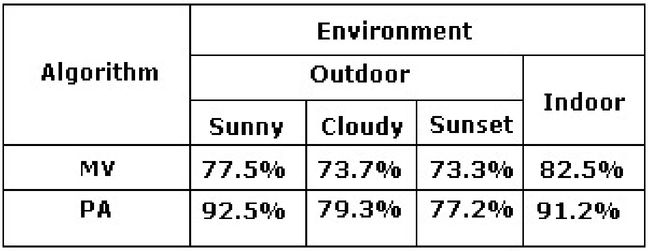

(a) (b) (c)

Fig. 4. Tracking of object in indoor location (a) Background (b) Object marked by user (c) Tracked object

Table 1. Comparison of accuracy



Algorithm	Environment			
	Outdoor			Indoor
	Sunny	Cloudy	Sunset	
MV	77.5%	73.7%	73.3%	82.5%
PA	92.5%	79.3%	77.2%	91.2%

paper and MV denotes a method based on motion vectors only. Performance evaluation was done "frame wise" in this study. This means, tracking output and ground truth were compared on a frame-by-frame basis. An object was considered "missed" in a frame by our system if less than 50% of it was tracked correctly. Results have been grouped together under various shooting conditions, like indoor, sunny, cloudy and sunset conditions and compared with a motion vector based tracking algorithm proposed in [11].

We also compared the time efficiency of our algorithm with the motion vector based prediction algorithm and derived that object tracking using background subtraction and motion estimation is much faster. A comparative performance in terms of speed is presented in Fig. 5 for a 1.8 GHz Pentium IV machine.

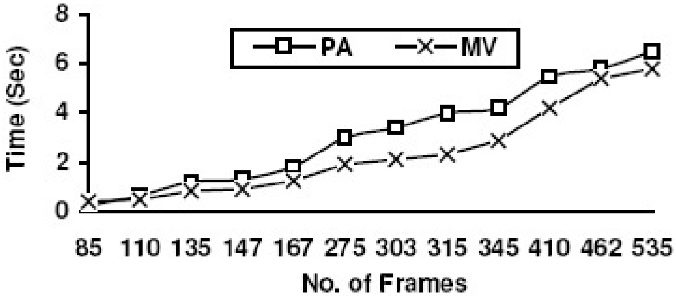


Fig. 5. Speed comparison with motion vector based tracking

As seen in the above figure, the combined approach is almost as efficient as a simple motion vector based approach, while its performance is much better as presented in Table 1. Another important observation is that, the proposed method can process frames at a rate of about 100 frames per second. Hence, it can be used in any real-time video tracking application.

4 Conclusion

We have proposed a novel tracking method that effectively tracks objects in a compressed video by combining background subtraction and motion estimation. The system consists of four main components: background subtraction, candidate object identification, target object selection and object interpolation. Although we work strictly in the compressed domain, we still get high levels of accuracy with minimal processing time and computational cost. Several issues may further be addressed. This includes handling of full occlusions, fast camera motion, multiple object tracking and unsupervised tracking of objects.

Acknowledgment

The work done by Shamik Sural is supported by research grants from the Department of Science and Technology, India, under Grant No. SR/FTP/ETA-20/2003 and by a grant from IIT Kharagpur under ISIRD scheme No. IIT/SRIC/ISIRD /2002-2003. Work done by A. K. Majumdar is supported by a research grant from the Department of Science and Technology, India, under Grant No. SR/S3/EECE /024/2003-SERC-Engg.

References

1. G. L. Foresti and F. Roli: Real-time Recognition of Suspicious Events for Advanced Visual-based Surveillance. In Multimedia Video-Based Surveillance Systems: From User Requirements to Research Solutions, G. L. Foresti, C. S. Regazzoni, and P. Mahonen, Eds. Dordrecht, The Netherlands: Kluwer, pp. 84 - 93, 2000.
2. Y. Wang, R.E. Van Dyke and J F Doherty: Tracking Moving Objects in video Scene. Technical Report, Department of Electrical Engg, Pennsylvania State University, 2000.
3. V.V. Vinod and H. Murase: Video Shot Analysis using Efficient Multiple Object Tracking. Proceedings of IEEE International Conference on Multimedia Computing and Systems, pp. 501 - 508, 1997.
4. L. Wixson: Detecting Salient Motion by Accumulating Directionally-Consistent Flow. IEEE Transactions on PAMI, Vol. 22, No. 8, pp. 774 - 780, 2000.
5. I.O. Sebe: Object-Tracking using Multiple Constraints. Technical Report, Department of Electrical Engg, Stanford University, 2002.
6. O.Sukmarg and K.R. Rao: Fast Object Detection and Segmentation in MPEG Compressed Domain. Proceedings of TENCON, Vol. 3, pp. 364 - 368, 2000.
7. V. Mezaris et al: Real-Time Compressed-Domain Spatiotemporal Segmentation and Ontologies for Video Indexing and Retrieval. IEEE Transactions on CSVT, Vol. 14, No 5, pp. 606 - 620, 2004.
8. S. M. Park and J. Lee: Tracking using Mean Shift Algorithm. Proceedings of International Conference of Information Communications and Signal Processing, 2003, pp 748 - 752, 2003.
9. W.Y. Yoo and J.Lee: Analysis of Camera Operations in Compressed Domain based on Generalized Hough Transform. Proceedings of IEEE Pacific Rim Conference on Multimedia, Vol. 2195, pp. 1102 - 1107, 2001.
10. H. Kartik, D. Schonfeld, P. Raffy, F. Yassa: Object Tracking Using Block Matching. IEEE conference on Image Processing, Vol. 3, pp. 945 - 948, Jul 2003.
11. M.Kankanhalli, R. Achanta and J. Wang: A Sensor Fusion Based Object Tracker for Compressed Video. Proceedings of the Sixth International Workshop on Advanced Image Technology, 2003.

Multi-camera Tracking of Articulated Human Motion Using Motion and Shape Cues

Aravind Sundaresan and Rama Chellappa

Department of Electrical and Computer Engineering,
University of Maryland, College Park, MD 20742, USA
{aravinds, rama}@cfar.umd.edu
http://www.cfar.umd.edu/users/aravinds/

Abstract. We present a framework and algorithm for tracking articulated motion for humans. We use multiple calibrated cameras and an articulated human shape model. Tracking is performed using motion cues as well as image-based cues (such as silhouettes and “motion residues” hereafter referred to as spatial cues,) as opposed to constructing a 3D volume image or visual hulls. Our algorithm consists of a predictor and corrector: the predictor estimates the pose at the $t + 1$ using motion information between images at t and $t + 1$. The error in the estimated pose is then corrected using spatial cues from images at $t + 1$. In our predictor, we use robust multi-scale parametric optimisation to estimate the pixel displacement for each body segment. We then use an iterative procedure to estimate the change in pose from the pixel displacement of points on the individual body segments. We present a method for fusing information from different spatial cues such as silhouettes and “motion residues” into a single energy function. We then express this energy function in terms of the pose parameters, and find the optimum pose for which the energy is minimised.

1 Introduction

The complex articulated structure of human beings makes tracking articulated human motion a difficult task. It is necessary to use multiple cameras to deal with occlusion and kinematic singularities. We also need shape models to deal with the large number of body segments and to exploit their articulated structure. In our work, we use shape models, whose parameters are known, to build a system that can track articulated human body motion using multiple cameras in a robust and accurate manner. A tracking system works better if there are more number of observations to estimate the pose and to that end our system uses different kinds of cues that can be estimated from the images. We use both motion information (in the form of pixel displacements), as well as spatial information (such as silhouettes, and “motion residues”, hereafter referred to as spatial cues). The motion and spatial cues are complementary in nature. We present a framework for unifying different spatial cues into a single energy image. The energy of a pose can be described in terms of this energy image. We can then obtain the pose that possesses the least energy using optimisation

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 131–140, 2006.

techniques. Much of the work in the past has focussed on using either motion or spatial parameters. In this paper we present an algorithm that fuses together information from these two kinds of cues. Since we use motion and spatial cues in our tracking algorithm, we are able to better deal with cases where the body segments are close to each other, such as when the arms are by the side of the body. Purely silhouette based methods typically experience difficulties in such cases. Silhouette or edge-based methods also have the weakness that they will not be able to deal with rotation about the axis of the body segment.

Estimating the initial pose is a different problem from tracking and is difficult due to the large number of unknown parameters (joint angles). It is computationally intensive and typically requires several additional algorithms such as head detectors or hand detectors. Stochastic algorithms such as particle filtering or optimisation methods are required for the sake of robustness. While the methods we present in this paper can be used for initialisation as well, we concentrate on the tracking aspect.

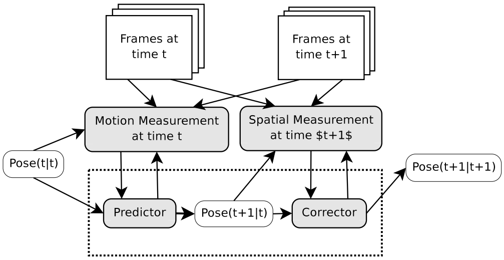
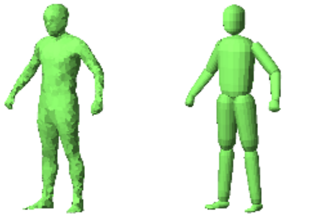


Fig. 1. Overview of the algorithm



(a) 3D Scan (b) Super-quadric

Fig. 2. 3D model comparison

In our work, we use eight cameras that are placed around the subject. We use parametric shape models connected in an articulated tree to represent the human body as described in Section 1.2.Our system, the block diagram of which is presented in Figure 1, consists of two parts: a predictor and corrector. We assume that the initial pose is known. The tracking algorithm is as follows.

- Compute 2D pixel displacement between frames at times t and $t + 1$.
- Predict 3D pose at $t + 1$ based on 2D motion from multiple cameras.
- Compute an energy function that fuses information from different spatial cues.
- Use the energy function to refine estimate of pose at $t + 1$.

We represent the pose, $\boldsymbol{\varphi}_t$, in a parametric form as a vector of the position of the base-body (6 degrees of freedom) and the joint angles of the various articulated body segments (3 degrees of freedom for each joint.) $\boldsymbol{\delta}$ represents the incremental pose vector.

We summarise prior work in articulated tracking in Section 1.1. We then describe the models in Section 1.2 and the details of our algorithm in Section 2.

We validate our algorithm using real images captured from eight cameras and the results are presented in Section 3.

1.1 Prior Work

We address the problem of tracking articulated human motion using multiple cameras. Gavrila and Davis [1], Aggarwal and Cai [2], and Moeslund and Granum [3], provide surveys of human motion tracking and analysis methods. a We look at some existing methods that use either motion-based methods or silhouette or edge based methods to perform tracking. Yamamoto and Koshikawa [4] analyse human motion based on a robot model and Yamamoto et al. [5] track human motion using multiple cameras. Gavrila and Davis [6] discuss a multi-view approach for 3D model-based tracking of humans in action. They use a generate-and-test algorithm in which they search for poses in a parameter space and match them using a variant of Chamfer matching. Bregler and Malik [7] use an orthographic camera model and use optical flow. Rehg and Morris [8] and Rehg et al. [9] describe ambiguities and singularities in tracking of articulated objects and Cham and Rehg [10] propose a 2D scaled prismatic model. Sidenbladh et al. [11] provide a framework to track 3D human figures using 2D image motion and particle filters with a constrained motion model that restricts the kinds of motions that can be tracked. Kakadiaris and Metaxas [12] use silhouettes from multiple cameras to estimate 3D motion. Plaenkers and Fua [13] use articulated soft objects with an articulated underlying skeleton as a model and use stereo and silhouette data for shape and motion recovery. Theobalt et al. [14] project the texture of the model obtained from silhouette-based methods and refine the pose using the flow field. Delamarre and Faugeras [15] use 3D articulated models for tracking with silhouettes. They use silhouette contours and apply forces to the contours obtained from the projection of the 3D model so that they move towards the silhouette contours obtained from multiple images. Cheung et al. [16] use shapes from silhouette to estimate human body kinematics. Chu et al. [17] use volume data to acquire and track a human body model. Wachter and Nagel [18] track persons in monocular image sequences. They use an IEKF with a constant motion model and use edges to region information in the pose update step in their work. Moeslund and Granum [19] use multiple cues for model-based human motion capture and use kinematic constraints to estimate pose of a human arm. The multiple cues are depth (obtained from a stereo rig) and the extracted silhouette, whereas the kinematic constraints are applied in order to restrict the parameter space in terms of impossible poses. Sigal et al. [20, 21] use non-parametric belief propagation to track in a multi view set up. Lan and Huttenlocher [22] use hidden Markov temporal models. DeMirdjian et al. [23] constrain pose vectors based on kinematic models using SVMs. Rohr [24] performs automated initialisation of the pose for single camera motion. Krahnstoever [25] addresses the issue of model acquisition and initialisation. Mikic et al. [26] automatically extract the model and pose using voxel data. Ramanan and Forsyth [27] also suggest an algorithm that performs rough pose estimation and can be used in an initialisation step. Sminchisescu and Triggs present a method for monocular video sequences using robust image matching, joint limits and non-self-intersection constraints [28]. They also try to remove kinematic ambiguities in monocular pose estimation efficiently [29].

Our method is different in that we use both motion and spatial cues to track the pose as opposed to using volume or visual based techniques or only optical flow. We use spatial and motion cues obtained from multiple views in order to obtain robust results that overcome occlusions and kinematic singularities. We also present a novel method to use spatial cues such as silhouettes and motion residues. It is also possible to incorporate edges in our method. We also do not constrain the motion or the pose parameters for specific types of motion (such as walking) and hence our method is general.

1.2 Models

A good human shape model should allow the system to represent the human body in all of it's postures and yet be simple enough to minimise the number of parameters required to represent the body accurately. We use tapered super-quadrics in order to represent the different body segments. We can use more complex triangular mesh models if we can acquire the parameters of such models. We illustrate the 3D model used in our experiments in Figure 2. The dimensions of the super-quadrics are obtained manually with the help of the 3D scanned model in the figure. The motion of the different body segments are constrained by the articulated structure of the body. The base body (trunk) has 6 degree-of-freedom (DoF) motion. All other body segments are attached to the base body in a kinematic chain and have at most 3 DoF rotational motion with respect to the parent node. The body model also includes the locations of the joints of the different body segments besides the shape of the body segment.

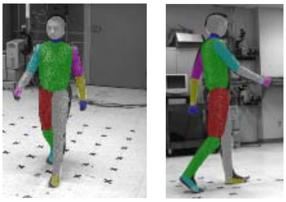
2 Algorithm

We compute the pose at time $t+1$ given the pose at time t using the images at time t and $t+1$. The pose at $t+1$ is estimated in two steps, the prediction step and the correction step. The steps required to estimate the pose at time $t+1$ are first listed and then described in detail in the sections that follow.

1. Pixel-body registration at time t using known pose at t.
2. Estimate pixel displacement between time t and time $t+1$.
3. Predict pose at time $t+1$ using pixel displacement.
4. Combine silhouettes and “motion residue” for each body segment into an “energy image” for each image.
5. Correct the predicted pose at time $t+1$ using the “energy image” obtained in step 4.

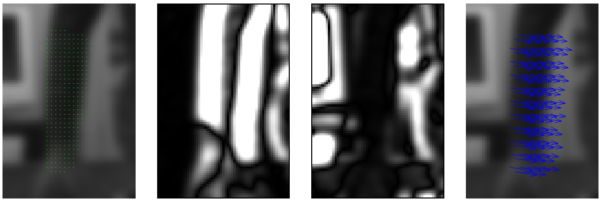
2.1 Pixel-Body Registration

Pixel-body registration is the process of registering each pixel in each image to a body segment as well as obtain approximate 3D coordinates of the point. We



(a) View 1 (b) View 2

Fig. 3. Pixel registration



(a) Mask (b) Image Diff (c) MR (d) Flow

Fig. 4. Pixel displacement and Motion Residue

thus obtain a 2D mask for each body segment that we can use while estimating the pixel displacement. We convert each body segment into a triangular mesh and project it onto each image, and compute the depth at each pixel by interpolating the depths of the triangle vertices. We can thus fairly easily extend our algorithm to use triangular mesh models instead of super-quadrics. Since the depths of all pixels are known, we can compute occlusions. Figure 3 illustrates the projection of the body onto images from two cameras. Different colours indicate different body segments. We compute approximate 3D coordinates of pixels in a similar fashion.

2.2 Estimating Pixel Displacement

As we use pixel displacement between frames to estimate 3D pose change, we are not dependent on specific optical flow algorithms. Figure 4 illustrates how we obtain the pixel displacement of a single body segment, the example being that of the left forearm shown in Figure 3 (d). We use a robust parametric model for the motion of the rigid objects so that the displacement, $\Delta\boldsymbol{x}_i$, at pixel $\boldsymbol{x}_i$ is given by $\Delta(\boldsymbol{x}_i, \boldsymbol{\phi})$, where $\boldsymbol{\phi} = [u, v, \theta, s]$. The elements of $\boldsymbol{\phi}$ are the displacements along the x and y axes, rotation and scale respectively. We find that the above parametric representation is more intuitive and more robust than an affine model. We obtain that value of $\boldsymbol{\phi} \in [\boldsymbol{\phi}_0 - \boldsymbol{\phi}_B, \boldsymbol{\phi}_0 + \boldsymbol{\phi}_B]$ that minimises the residue given by $\boldsymbol{e}^\mathsf{T}\boldsymbol{e}$ where

$$[\boldsymbol{e}]_j = I_t(\boldsymbol{x}_{i_j}) - I_{t+1}(\boldsymbol{x}_{i_j} + \Delta(\boldsymbol{x}_{i_j}, \boldsymbol{\phi})),$$

and $\{\boldsymbol{x}_{i_j} : j = 1, 2, \cdots\}$ is the set of all points in the mask obtained in Section 2.1 and illustrated in Figure 4 (a). $\boldsymbol{\phi}$ denotes zero motion and $\boldsymbol{\phi}_B$ denotes the bounds on the motion that we impose. Figure 4 (a) is the smoothed intensity image at time t. Figure 4 (b) is the difference between image at time t and $t + 1$, i.e., with zero motion, and has large values in the mask region signifying that there is some motion. Figure 4 (b) is the difference between image at time t and the image at time $t + 1$ warped according to the estimated motion and is called the "motion residue" for the optimal $\boldsymbol{\phi}$. The value of the pixels in the region of the mask is close to zero where the estimated pixel displacement agrees with the actual pixel displacement. The "motion residue" provides us with a rough delineation of the location of the body segment, even when the original mask does not exactly match the body segment.

2.3 Pose Prediction

The pose parameter we need to estimate is the vector $\boldsymbol{\varphi}$, which consists of the 6-DoF parameters for the base-body and the 3-DoF joint angles for each of the remaining body segments. The state vector in our state-space formulation is $\boldsymbol{\varphi}_t$ (1-2).

$$\text{State Update :} \qquad \boldsymbol{\varphi}_{t+1} = \boldsymbol{h}(\boldsymbol{\varphi}_t) + \boldsymbol{\delta}_t \tag{1}$$

$$\text{Observation :} \qquad \boldsymbol{f}(\Delta\boldsymbol{x}_t, \boldsymbol{\varphi}_t, \boldsymbol{\varphi}_{t+1}) = 0 \tag{2}$$

In our case the function $\boldsymbol{h}(.)$ is linear (3) and the pixel position $\boldsymbol{x}(.)$, in (4), is a non-linear function of the pose, $\boldsymbol{\varphi}$, and the incremental pose, $\boldsymbol{\delta}$. However, it is well approximated by a linear function locally.

$$\boldsymbol{\varphi}_{t+1} = \boldsymbol{\varphi}_t + \boldsymbol{\delta}_t \tag{3}$$

$$\boldsymbol{f}(\Delta\boldsymbol{x}_t, \boldsymbol{\varphi}_t, \boldsymbol{\delta}_t) = \Delta\boldsymbol{x}_t - (\boldsymbol{x}(\boldsymbol{\varphi}_t + \boldsymbol{\delta}_t) - \boldsymbol{x}(\boldsymbol{\varphi}_t)) \tag{4}$$

Let us consider the observation, the measured (noisy) pixel displacement, $\Delta\boldsymbol{x}'_t = \Delta\boldsymbol{x}_t + \boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is the measurement noise, and $\Delta\boldsymbol{x}_t$ is the pixel displacement. We expand $\boldsymbol{f}(\Delta\boldsymbol{x}_t, \boldsymbol{\varphi}_t, \boldsymbol{\delta}_t)$ in a Taylor series about $\boldsymbol{f}(\Delta\boldsymbol{x}'_t, \hat{\boldsymbol{\varphi}}_t, \hat{\boldsymbol{\delta}}_t)$ as

$$\boldsymbol{f}\left(\Delta\boldsymbol{x}'_t, \hat{\boldsymbol{\varphi}}_t, \hat{\boldsymbol{\delta}}_t\right) + \frac{\partial \boldsymbol{f}}{\partial \Delta\boldsymbol{x}_t}(\Delta\boldsymbol{x}_t - \Delta\boldsymbol{x}'_t) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\varphi}_t}(\boldsymbol{\varphi}_t - \hat{\boldsymbol{\varphi}}_t) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\delta}_t}\left(\boldsymbol{\delta}_t - \hat{\boldsymbol{\delta}}_t\right) + \boldsymbol{\mathcal{O}}(\cdots). \tag{5}$$

The left hand side $(\boldsymbol{f}(\Delta\boldsymbol{x}_t, \boldsymbol{\varphi}_t, \boldsymbol{\delta}_t))$ is 0. The first term $\boldsymbol{f}\left(\Delta\boldsymbol{x}'_t, \hat{\boldsymbol{\varphi}}_t, \hat{\boldsymbol{\delta}}_t\right)$ is given by $\Delta\boldsymbol{x}'_t - \left(\boldsymbol{x}(\hat{\boldsymbol{\varphi}}_t + \hat{\boldsymbol{\delta}}_t) - \boldsymbol{x}(\hat{\boldsymbol{\varphi}}_t)\right)$.

The second term can be simplified as $\frac{\partial \boldsymbol{f}}{\partial \Delta\boldsymbol{x}_t}(\Delta\boldsymbol{x}_t - \Delta\boldsymbol{x}'_t) = 1.(-\boldsymbol{\eta}) = -\boldsymbol{\eta}$. The third term in (5) $\frac{\partial \boldsymbol{f}(.)}{\partial \boldsymbol{\varphi}_t}(\boldsymbol{\varphi}_t - \hat{\boldsymbol{\varphi}}_t)$ is negligible because the function $\boldsymbol{f}(.)$ is not very sensitive to the current pose, $\boldsymbol{\varphi}_t$ and we expect the term $\boldsymbol{\varphi}_t - \hat{\boldsymbol{\varphi}}_t$ to be also negligible. We assume, without loss of generality that $\boldsymbol{\delta}_t$ is a linear function of time t, so that $\boldsymbol{\delta}_t = \boldsymbol{\delta}.t$, where $\boldsymbol{\delta}$ is a constant. We note that (6) follows from the fact that the pixel velocity, $\frac{\partial \boldsymbol{x}(\boldsymbol{\varphi}_t)}{\partial t}$, at a given point is a linear function of the rate of change of pose, $\boldsymbol{\delta}$ [30].

$$\frac{\partial \boldsymbol{f}\,(\Delta\boldsymbol{x}, \boldsymbol{\varphi}, \boldsymbol{\delta}_t)}{\partial \boldsymbol{\delta}_t} = -\frac{\partial \boldsymbol{x}(\boldsymbol{\varphi} + \boldsymbol{\delta}_t)}{\partial t} / \frac{\partial \boldsymbol{\delta}_t}{\partial t} = -F\,(\boldsymbol{\varphi} + \boldsymbol{\delta}_t)\,\boldsymbol{\delta}/\boldsymbol{\delta} = -F\,(\boldsymbol{\varphi}_t + \boldsymbol{\delta}_t) \tag{6}$$

The fourth term is $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\delta}_t} \mid_{(\Delta\boldsymbol{x}'_t, \hat{\boldsymbol{\varphi}}_t, \hat{\boldsymbol{\delta}}_t)} = -F\left(\hat{\boldsymbol{\varphi}}_t + \hat{\boldsymbol{\delta}}_t\right)$ We neglect the higher order terms in (5) and obtain the following linearised observation equation (7).

$$\Delta\boldsymbol{x}'_t - \left(\boldsymbol{x}(\hat{\boldsymbol{\varphi}}_t + \hat{\boldsymbol{\delta}}_t) - \boldsymbol{x}(\hat{\boldsymbol{\varphi}}_t)\right) + \boldsymbol{\eta} = F\left(\hat{\boldsymbol{\varphi}}_t + \hat{\boldsymbol{\delta}}_t\right)\left(\boldsymbol{\delta}_t - \hat{\boldsymbol{\delta}}_t\right) \tag{7}$$

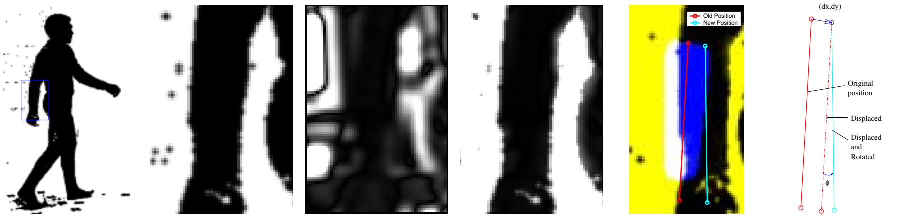
We solve (7) for $\boldsymbol{\delta}_t$ iteratively. We set $\hat{\boldsymbol{\delta}}_t^0 = 0$ and perform the following until we obtain numerical convergence, which we do in a few iterations. We finally set $\hat{\boldsymbol{\varphi}}_{t+1} = \hat{\boldsymbol{\varphi}}_t + \hat{\boldsymbol{\delta}}_t^N$.

- Set $F^{(i)} = F\left(\hat{\boldsymbol{\varphi}}_t + \hat{\boldsymbol{\delta}}_t^{(i)}\right)$.
- Set $\Delta\boldsymbol{x}_t^{(i)} = \Delta\boldsymbol{x}'_t - \left(\boldsymbol{x}\left(\hat{\boldsymbol{\varphi}}_t + \hat{\boldsymbol{\delta}}_t^{(i)}\right) - \boldsymbol{x}(\hat{\boldsymbol{\varphi}}_t)\right)$
- Update pose: $\hat{\boldsymbol{\delta}}_t^{(i+1)} = \hat{\boldsymbol{\delta}}_t^{(i)} + \left(F^{(i)T}F^{(i)}\right)^{-1} F^{(i)T}\Delta\boldsymbol{x}_t^{(i)}$.

2.4 Computing Spatial Energy Function

We combine different types of spatial cues into an energy image for each body segment. This allows us to use the framework irrespective of which spatial cues are available. In our work we use silhouette information as well as the "motion residue" obtained during motion estimation.

Figure 4 (d) is the "motion residue" for that segment, and provides us with the region that agrees with the motion of the mask. We combine the "motion residue" with the silhouette as shown in Figure 5. We can form energy images even if the quality of the silhouette is not very good. There are a number of outliers, but though these may affect other silhouette based algorithms, they do not affect our algorithm much.



(a) Silhouette (b) Silhouette(c) Motion Residue (d) Energy (e) Object mask (f) 2D pose

Fig. 5. Obtaining unified energy image for the forearm

Once we have the pixel-wise energy image for each camera and a given body segment we compute the energy for different values of 2D parameters such as displacement and rotation. We have a mask for the body segment for the body segment for a given image as illustrated in Figure 5 (e). We can move this mask by a translation (dx, dy) or a rotation φ as illustrated in Figure 5 (f). We can find the "energy" of the mask in each position by summing the energy of all the pixels that belong to the mask. Thus we can express the energy as a function of (dx, dy, θ) in the neighbourhood of $(dx, dy, \theta) = (0, 0, 0)$. When the body segment moves in 3D space by a translation and rotation, we can project the new axis on to the image and find the corresponding 2D configuration parameters in each of the images. We can then find the energy of the 3D pose by summing the energies of the mask in the 2D configurations in each image.

We minimise this energy function in the local neighbourhood. We use a Levenberg-Marquardt optimisation technique which is initialised to the current 3D position. We show the new position of the axis of the body segment after optimisation in Figure 6. The red line represents the initial position of the axis of the body segment and the cyan line represents the new position. We thus correct the pose using spatial cues.

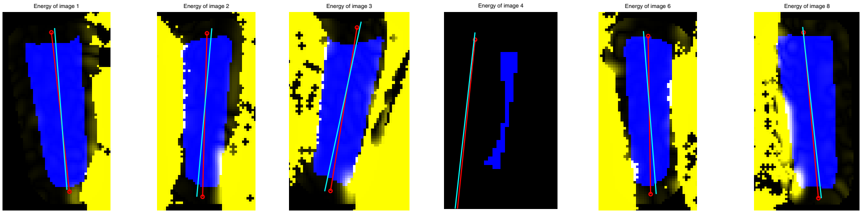


Fig. 6. Minimum energy configuration

3 Experimental Results and Conclusions

In the experiments performed, we use grey-scale images from eight cameras with a spatial resolution of 648 × 484. Calibration is performed using Tomas Svoboda's algorithm [31] and a simple calibration device to compute the scale. We use images that have been undistorted based on the radial calibration parameters of the cameras. We use perspective projection model for the cameras. Experiments were conducted on different kind of sequences and we present the results of two such experiments. The subject performs motions that exercise several joint angles in the body. Our results show that using only motion cues for tracking causes the pose estimator to lose track eventually, as we are estimating only the *difference in the pose* and therefore the error accumulates. This underlines the need for "correcting" the pose estimated using motion cues. We show the "correction" step of the algorithm prevents drift in the tracking. In Figure 7, we present results in which we have superimposed the images with the model assuming the estimated pose over the images obtained from two cameras. The length of the first sequence is 10 seconds (300 frames), during which there is considerable movement and bending of the arms and occlusions at various times in different cameras. The second sequence is that of the subject walking and the body parts are successfully tracked in both cases.

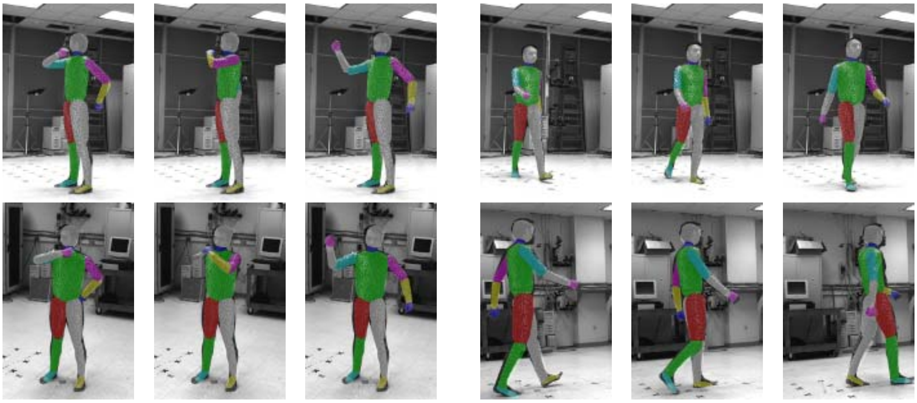


Fig. 7. Tracking results using both motion and spatial cues

We note that the method is fairly accurate and robust despite the fact the human body model used is not very accurate, given that it was obtained manually using visual feedback. Specifically, the method is sensitive to joint location and it is important to accurately estimate the joint location during the model acquisition stage. We also note that the method scales with respect to accuracy of the human body model. We also note that while we use super-quadrics to represent body segments, we could easily use triangular meshes instead, provided they can be obtained. We need to consider more flexible models that allow the location of certain joints, such as shoulder joints, to vary with respect to the trunk, to better model the human body.

References

1. Gavrila, D.M.: The visual analysis of human movement: A survey. Computer Vision and Image Understanding: CVIU **73** (1999) 82–98
2. Aggarwal, J., Cai, Q.: Human motion analysis: A review. Computer Vision and Image Understanding **73** (1999) 428–440
3. Moeslund, T., Granum, E.: A survey of computer vision-based human motion capture. CVIU (2001) 231–268
4. Yamamoto, M., Koshikawa, K.: Human motion analysis based on a robot arm model. In: CVPR. (1991) 664–665
5. Yamamoto, M., Sato, A., Kawada, S., Kondo, T., Osaki, Y.: Incremental tracking of human actions from multiple views. In: CVPR. (1998) 2–7
6. Gavrila, D., Davis, L.: 3-D model-based tracking of humans in action: A multi-view approach. In: Computer Vision and Pattern Recognition. (1996) 73–80
7. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. In: CVPR. (1998) 8–15
8. Rehg, J.M., Morris, D.: Singularity analysis for articulated object tracking. In: Computer Vision and Pattern Recognition. (1998) 289–296
9. Rehg, J., Morris, D.D., Kanade, T.: Ambiguities in visual tracking of articulated objects using two- and three-dimensional models. International Journal of Robotics Research **22** (2003) 393 – 418
10. Cham, T.J., Rehg, J.M.: A multiple hypothesis approach to figure tracking. In: Computer Vision and Pattern Recognition. Volume 2. (1999)
11. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3D human figures using 2D image motion. In: ECCV. (2000) 702–718
12. Kakadiaris, I., Metaxas, D.: Model-based estimation of 3D human motion. IEEE PAMI **22** (2000) 1453–1459
13. Plänkers, R., Fua, P.: Articulated soft objects for video-based body modeling. In: ICCV. (2001) 394–401
14. Theobalt, C., Carranza, J., Magnor, M.A., Seidel, H.P.: Combining 3D flow fields with silhouette-based human motion capture for immersive video. Graph. Models **66** (2004) 333–351
15. Delamarre, Q., Faugeras, O.: 3D articulated models and multi-view tracking with silhouettes. In: ICCV. (1999) 716–721
16. K.M. Cheung, S.B., Kanade, T.: Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In: IEEE Conference on Computer Vision and Pattern Recognition. (2003) 77–84

17. Chu, C.W., Jenkins, O.C., Mataric, M.J.: Markerless kinematic model and motion capture from volume sequences. In: CVPR (2). (2003) 475–482
18. Wachter, S., Nagel, H.H.: Tracking persons in monocular image sequences. Computer Vision and Image Understanding **74** (1999) 174–192
19. Moeslund, T., Granum, E.: Multiple cues used in model-based human motion capture. In: International Conference on Face and Gesture Recognition. (2000)
20. Sigal, L., Isard, M., Sigelman, B.H., Black, M.J.: Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In: NIPS. (2003)
21. Sigal, L., Bhatia, S., Roth, S., Black, M.J., Isard, M.: Tracking loose-limbed people. In: CVPR. (2004) 421–428
22. Lan, X., Huttenlocher, D.P.: A unified spatio-temporal articulated model for tracking. In: CVPR (1). (2004) 722–729
23. Demirdjian, D., Ko, T., Darrell, T.: Constraining human body tracking. In: ICCV. (2003) 1071–1078
24. Rohr, K.: Human Movement Analysis Based on Explicit Motion Models. Kluwer Academic (1997)
25. Krahnstoever, N., Sharma, R.: Articulated models from video. In: Computer Vision and Pattern Recognition. (2004) 894–901
26. Mikic, I., Trivedi, M., Hunter, E., Cosman, P.: Human body model acquisition and tracking using voxel data. International Journal of Computer Vision **53** (2003) 199–223
27. Ramanan, D., Forsyth, D.A.: Finding and tracking people from the bottom up. In: CVPR (2). (2003) 467–474
28. Sminchisescu, C., Triggs, B.: Covariance scaled sampling for monocular 3D body tracking. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA. Volume 1. (2001) 447–454
29. Sminchisescu, C., Triggs, B.: Kinematic jump processes for monocular 3D human tracking. In: International Conference on Computer Vision & Pattern Recognition. (2003) I 69–76
30. Sundaresan, A., RoyChowdhury, A., Chellappa, R.: Multiple view tracking of human motion modelled by kinematic chains. In: International Conference on Image Processing, Singapore. (2004)
31. Svoboda, T., Martinec, D., Pajdla, T.: A convenient multi-camera self-calibration for virtual environments. PRESENCE: Teleoperators and Virtual Environments **14** (2005) To appear.

Matching Gait Image Sequences in the Frequency Domain for Tracking People at a Distance

Ryusuke Sagawa, Yasushi Makihara, Tomio Echigo, and Yasushi Yagi

Institute of Scientific and Industrial Research, Osaka University,
8-1 Mihogaoka, Ibaraki-shi, Osaka, 567-0047, Japan
{sagawa, echigo, yagi}@am.sanken.osaka-u.ac.jp

Abstract. This paper describes a new method to track people walking by matching their gait image sequences in the frequency domain. When a person walks at a distance from a camera, that person often appears and disappears due to being occluded by other people and/or objects, or by going out of the field of view. Therefore, it is important to track the person by taking correspondence of the image sequences between before and after the disappearance. In the case of tracking, the computational time is more crucial factor than that in the case of identification. We create a three-dimensional volume by piling up an image sequence of human walking. After using Fourier analysis to extract the frequency characteristics of the volume, our method computes the similarity of two volumes. We propose a method to compute their correlation of the amplitude of the principal frequencies to improve the cost of comparison. Finally, we experimentally test our method and validate that the amplitude of principal frequencies and spatial information are important to discriminate gait image sequences.

1 Introduction

When a surveillance system using cameras tracks people who walk at a distance from the cameras, the subjects are often occluded by other people and/or objects. Therefore, it is necessary to make correspondences of tracked people between before and after their occlusion. For example, if a multiple-camera system tracks people as shown in Figure 1, making correspondences of objects is necessary between before and after the occluded area. Since the images of people at a distance from a camera are small, it is difficult to recognize them by their facial appearance. Though color and shape are considered cues for matching, this paper focuses on the gaits of people, which are also important features. When walking, people move their torso, arms, and legs in a unique way. Hence the rhythm of a gait is different among individuals. Since gait can be observed at a distance, gait matching has advantages for a tracking system.

The issue of matching gait image sequence for tracking is similar to the identification problem of gait image sequence. However, the differences from identification are as follows:

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 141–150, 2006.

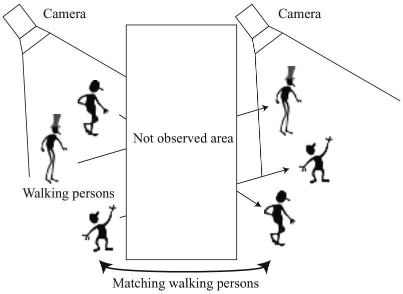


Fig. 1. Making correspondences of pedestrians with a multiple-camera system

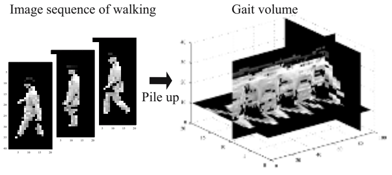


Fig. 2. A gait volume is created by piling up the image sequences of walking

- The features should be extracted only from an image sequence.
- The computational time is very important.

In the case of tracking problem, the features of walking should be extracted only from an image sequence while multiple image sequences can be used from a gait database in the case of identification. Moreover, the computational time is more crucial factor than that in the case of identification.

Several approaches have been proposed for identification of a person from their gait. They can mostly be classified into two classes, model- or appearance-based approaches. Model-based approaches extract the motion of the human body by fitting their models to input images. Yam et al. [1] and Cunado et al. [2] extracted leg motions and found their gait signature by Fourier analysis. Urtasun and Fua [3] used a 3D temporal motion model to increase the robustness for a changing view direction. Bobick and Johnson [4] extracted activity-specific static body parameters instead of directly analyzing gait motion. Lee and Grimson [5] analysed the frequency fo 7 parts which are extracted from the silhouette of human walking motion.

Appearance-based approaches directly extract parameters from images without assuming a model of a human body and its motion. Niyogi and Adelson [6] used 3D spatio-temporal (XYT) data by piling up images and extracted gait motion by fitting a 'snake'. Murase and Sakai [7] proposed a template matching method in the parametric eigenspace that is projected from images. Little and Boyd [8] recognized individuals by frequencies and phases computed by extracting optical flows. Liu and Picard [9] used a spatio-temporal volume and detected the periodicity in a motion by 1D Fourier analysis for each pixel of the image. BenAbdelkader et al. [10] used self-similarity plots, in which each pixel had correlations with the frames of an image sequence. Liu et al. [11] used a frieze pattern to represent gait motion; a pattern created by summing up the white pixels of a binarized image of a gait along the rows and columns of an image. Sarker et al. [12] proposed a baseline algorithm of gait recognition, which computes the similarity of gait sequences by spatial-temporal correlation. Han and Bhanu [13] proposed a representation of gait image sequence, called a gait energy image, which is computed by taking average of a silhouette image sequence.

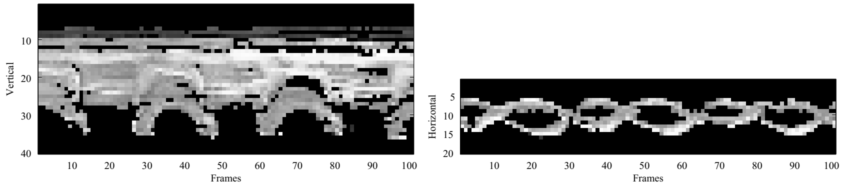
The previous model-based approaches to determine the frequency of a gait only considered the frequency of some parts of the body by extracting them from image sequence. However, we think that individuality is represented by a mixture of frequencies of whole body. Therefore, we attempted to extract the distribution of the various frequencies included in the motion of every part of a walking body. Our proposed approach creates 3D spatio-temporal volume from an image sequence, which is similar to Niyogi's [6] and Liu's [9] methods. Spatio-temporal volume data, here called *gait volume*, contain information not only of spatial individuality such as features of the torso and face, but also the movement of the body with its unique rhythm. By extracting the frequency characteristics of the volume by computing 1-D Fourier transform along a time axis, our method computes the similarity of two volumes. Though we analyze it by 3D Fourier transform in our previous paper [14], the spatial information is omitted. Thus, a new method utilizes the spatial information for matching. We propose a method to compute their correlation of the amplitude. Since it is not necessary for this method to align frames for matching two sequences, this method has advantage with respect to the computational cost.

In the following sections, we describe details of our method. In Section 2, we explain how a gait volume is created. In Section 3, we describe a method to extract frequency information by 1-D Fourier transform. Next, in Section 4, our method of matching frequency information is described. We experimentally test the proposed method in Section 5 and summarize our contribution in Section 6.

2 Creating a Spatio-temporal Volume

A gait volume is created by piling up image sequences of a person walking as shown in Figure 2. The process consists of two steps: background subtraction and image alignment. The human region is extracted by subtracting background from input images [15]. Moving regions are extracted as the human region by subtracting the stationary background images from input ones by pixels. After extracting the human region, the principle axis of the body is calculated as a horizontal position in the human region in an image if it is assumed that a person walks in a fronto-parallel plane. A sequence of the extracted human region is then temporally aligned by shifting the extracted human region. Without specifying each of the parts of the body, we can create a gait volume that contains the continuous changes of appearance while walking. These changes exactly reflect the gait rhythm.

A gait volume includes both spatial and temporal information. Sliced planes of the volume data express changes of textures in the subject's walking, and also represent the rhythm in a person's gait. Figure 3 shows examples of vertical and horizontal slices of a gait volume. From the slices, it is possible to acquire information about how a person moves his/her body while walking. Figure 3(a) is a vertical slice at the central column. There are vertical waves around the shoulder, arms and waist. Figure 3(b) is a horizontal slice at the row of the knee



(a) (b)

Fig. 3. (a) Vertical and (b) horizontal slices of a gait volume

level. There are horizontal waves of legs and the difference in the motions of the right and left legs can be observed.

3 Frequency Analysis of Gait Volume

In this section, we extract the frequency characteristics of a gait volume by Fourier transform. Our method consists of three steps:

1. Compute Fourier transform $G(x, y, k)$ of a gait volume $g(x, y, n)$ along the time axis.
2. Extract the principal frequencies of a gait volume.
3. Remove spectra from $G(x, y, k)$ other than the principal frequencies.

First, we compute 1-D discrete Fourier transform for each pixel of images along the frame axis:

$$G(x, y, k) = \sum_{n=0}^{N-1} g(x, y, n) \exp(-\frac{2\pi ikn}{N}), \tag{1}$$

where $g(x, y, n)$ is the intensity of a pixel (x, y) at n-th frame and N is the number of frames. Figure 4 shows an example of the change of the intensity of a pixel in a gait volume. After Fourier transform, the amplitude of the pixel becomes as shown in Figure 5. The frequency f corresponding to k is computed as $f = \frac{k}{N\Delta t}$, where Δt is the sampling interval of images.

Second, since the amplitudes of the most of frequencies are small while the dimension of $G(x, y, k)$ is very high, we extract the principal frequencies of a gait, which have large amplitudes, to reduce the data size and improve the computational cost. We compute the sum of the amplitude of $G(x, y, k)$ for each frequency:

$$\hat{G}(k) = \sum_{x,y} |G(x, y, k)|. \tag{2}$$

Since $G(x, y, k)$ is a complex value, $|G(x, y, k)| = \sqrt{a^2 + b^2}$, where $G(x, y, k) = a + bi$. Then, we find the principal frequency of $G(x, y, k)$ as the frequency k that satisfies $\hat{G}(k-1) < \hat{G}(k)$ and $\hat{G}(k+1) < \hat{G}(k)$. Since the higher frequencies is not important, we choose some lower frequencies from them. The DC component

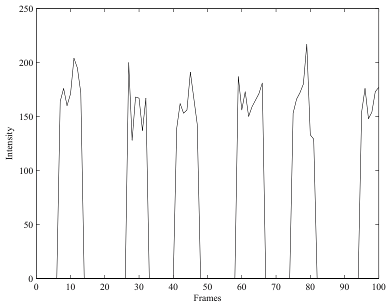


Fig. 4. Change of the intensity of a pixel in a gait volume

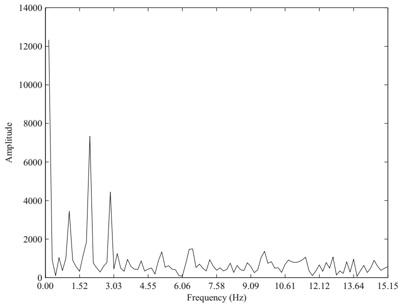


Fig. 5. Amplitude of a pixel after Fourier transform

is ignored for this computation because it does not represent a repetitive motion of walking. Figure 6 shows an example of $\hat{G}(k)$. There are some peaks and we extract their frequencies as the principal frequencies.

In the third step, we remove spectra included in $G(x, y, k)$ other than the principal frequencies and obtain a new volume $G'(x, y, k)$. We preserve the several lowest principal frequencies and remove all other frequencies. Figure 7 shows the amplitude of $G'(x, y, k)$ after removing spectra other than the principal frequencies from Figure 5.

Figure 8 shows the reconstructed results of the following inverse Fourier transform:

$$g'(x, y, n) = \frac{1}{N} \sum_{k=0}^{N-1} G'(x, y, k) \exp(\frac{2\pi ikn}{N}). \tag{3}$$

Figure 8(a) is an original image in a gait volume and Figure 8(b) is the reconstructed image from $G'(x, y, k)$. Figure 8(c) shows the amplitudes of three principal frequencies of $G'(x, y, k)$. Figure 8(d) is the horizontal slice of $g'(x, y, n)$, which corresponds to Figure 3(b).

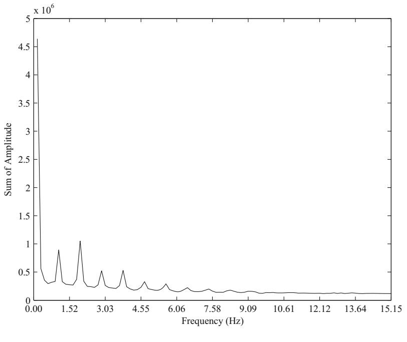


Fig. 6. Sum of the amplitude of $G(x, y, k)$

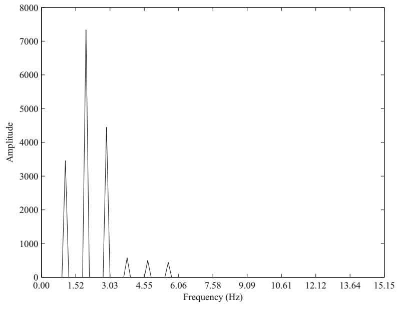


Fig. 7. Example of amplitudes of principal frequencies

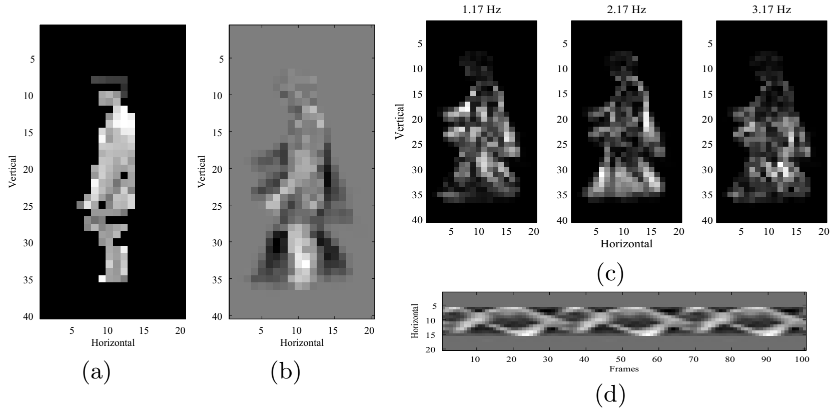


Fig. 8. (a) Original image, (b) reconstructed image by inverse Fourier transform, (c) Amplitudes of three principal frequencies, and (d) the horizontal slice of the reconstructed gait volume which corresponds to Figure 3(b)

4 Matching Gait Volumes

To compare two different gait volumes, we propose a method for computing their correlation. We use the correlation of the amplitude after Fourier transform. In this section, we assume that the images of walking people are aligned along the horizontal and vertical axis of gait volumes.

Now, $G_1(x, y, k)$ and $G_2(x, y, k)$ are the volumes after Fourier transform. If G_1 is a reference volume, we remove spectra other than the principal frequencies of G_1 from both G_1 and G_2, and obtain $G'_1(x, y, k)$ and $G'_2(x, y, k)$. Namely, G'_1 and G'_2 only have the principal frequencies of G_1. We compare the amplitude of G'_1 and G'_2 by the following criterion:

$$C(|G'_1|, |G'_2|) = \frac{\sum_{x,y,k} |G'_1(x, y, k)||G'_2(x, y, k)|}{\sqrt{(\sum_{x,y,k} |G'_1(x, y, k)|^2)(\sum_{x,y,k} |G'_2(x, y, k))|^2}}. \tag{4}$$

This is the normalized correlation without shifting by mean value. Thus, if two volumes are same, the result is 1, and it goes down to -1 if they have negative correlation.

Removing spectra other than the principal frequencies is equal to reducing the dimension of the component in gait volumes. Therefore, the cost of computing (4) is much smaller than that of computing the correlation of the original volumes by $C(|G_1|, |G_2|)$.

5 Experiments

We first tested our method with image sequences in which people walk in a fronto-parallel plane to the camera. We used 50 sequences of 9 persons, i.e., 4-7 sequences for each person. Each sequence consists of 200 frames, captured

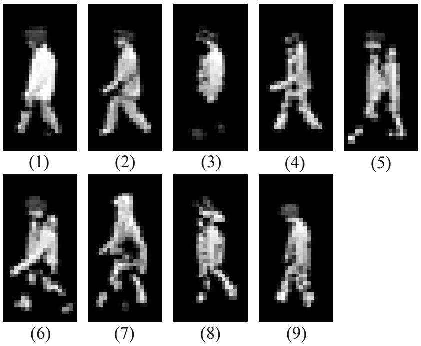


Fig. 9. Subject images

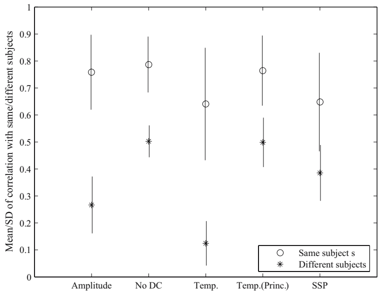


Fig. 10. Comparison of five methods: the means of comparing the same subjects are indicated by ∘, while those of comparing different subjects are indicated by *. Vertical lines show $\pm\sigma$, which is the standard deviation.

at 33Hz, and include 10-12 walking steps. The size of the images is 40 × 20. Figure 9 shows images of the subjects after background subtraction.

We compute the correlation of all pairs of sequences. To evaluate the efficiency of our method, we compared the following five methods:

- Amplitude: the correlation of $|G'(x, y, k)|$ proposed in Section 4.
- No DC: the correlation of $|G(x, y, k)|$ by (4) simply after removing the DC component.
- Temp.: normalized correlation in the spatio-temporal domain of $g(x, y, n)$.
- Temp.(Princ.): normalized correlation in the spatio-temporal domain of $g'(x, y, n)$, which is obtained by inverse Fourier transform of $G'(x, y, k)$.
- SSP: normalized correlation of the self-similarity plots proposed in [10].

In this experiment, we used the three lowest principal frequencies for matching. Thus, the data size is reduced to 3% of the original gait volume. Since a self-similarity plot is a matrix of the correlation of two images in a image sequence, the spatial information is lost in this representation. In the SSP method, we compute the normalized correlation of the self-similarity plots generated from gait sequences. Though the method of comparing the self-similarity plots is different from the one proposed in [10], we compare the effectiveness as a cue for identification by normalized correlation. In the Temp., Temp.(Princ.) and SSP methods, we search for the best match by an exhaustive brute-force search with circular shift in the spatio-temporal domain.

We apply the five above methods to all pairs of gait sequences. Figure 10 shows the mean and standard deviations of the correlations of each method when they are compared to the sequences of the same and different subjects. The means of comparing the same subjects are indicated by ∘ while those of comparing different subjects are indicated by *. The vertical lines shows $\pm\sigma$, which is the standard deviation.

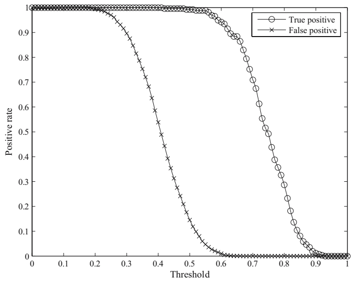


Fig. 11. Rates of positive samples after thresholding by correlation values

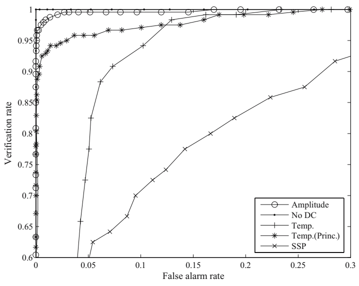


Fig. 12. ROC curves for five methods

If the difference between ○ and * is large, the method is effective to distinguish the gaits of different persons. In the Amplitude method, the difference is sufficiently large compared to their standard deviations. Therefore, the method can be seen as effective for comparing gait image sequences. Since the difference of the SSP method is small while the standard deviation is large, the robustness for matching is worse than others. Therefore, the spatial information in an image is important even if the frequency information is used for matching.

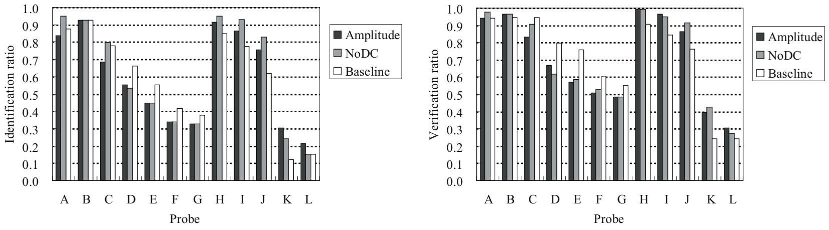
Figure 11 shows the rate of positive samples after thresholding by correlation values for the Amplitude method. The true positive is the result of matching the same subjects, and the false positive is that of matching different subjects. When the rate of true positive is 0.95, that of false positive is less than 0.01. Thus, the Amplitude method can discriminate subjects by thresholding the correlation values.

We evaluate the rates of positive samples for five methods, and create the receiver operating characteristic (ROC) curves as shown in Figure 12. It depicts the relationship of true and false positive rates. The No DC method has no error in this experiment. Therefore, it is shown that the frequency information is a powerful feature for matching. Though the data size for the Amplitude method is quite smaller than the No DC method, it has few errors. Hence, it is effective to use the principal frequencies for matching gait volumes. As for the Temp. and Temp.(Princ.) method, the result is worse than the No DC and Amplitude method. It shows that the template matching in the temporal domain is not suitable for matching gait volumes.

Table 1 shows the computational time for comparing a pair of gait sequences by these five methods. We used a PC with Pentium4 3.2GHz processor and coded the algorithms by MATLAB. The time of the Amplitude method is 16% of the

Table 1. Times for comparing a pair of gait sequences in seconds

Amplitude	No DC	Temp.	Temp.(Princ.)	SSP
0.015	0.093	4.0	4.0	2.5



(a) (b)

Fig. 13. Comparison of Amplitude, No DC and Baseline methods for USF database: (a) the identification rate at rank 5, (b) the verification rate at a false alarm rate of 10%. The differences between gallery and probe sequence: (A) view, (B) shoe, (C) shoe, view, (D) surface, (E) surface, shoe, (F) surface, view, (G) surface, shoe, view, (H) briefcase, (I) shoe, briefcase, (J) view, briefcase, (K) time, shoe, clothing, (L) surface, time, shoe, clothing.

No DC method. Thus, the computational cost is reduced by removing the minor component in $G(x, y, k)$. Since the temporal alignment is necessary for the other methods, their computational cost is higher than that of the Amplitude method.

Second, we tested our method by using a database of gait image sequence from University of South Florida; for details of the database, refer to [12]. The database consists of gallery (watch-list) and probe (input data) image sequences, which are compared in the experiment. We used the silhouettes which are already extracted by their algorithm in this experiment. The number of gallery sequences is 121. The size of images we use is normalized to 88 × 128 pixels, and the number of frames for matching is 128. We compared three methods, Amplitude, No DC and Baseline [12]. Figure 13 shows identification and verification rates for each probe. The difference between Amplitude and No DC methods is small while the cost of Amplitude method is much smaller than No DC method. Moreover, the costs of these methods are much smaller than Baseline method because aligning frames is necessary for Baseline method. Though the performance of Amplitude and No DC methods becomes worse than Baseline method for (B)-(G) probes, which have difference about surface, it is considered to be due to background subtraction. On the other hand, our method has advantage for (H)-(J) probes, which have difference about one's belongings. It is considered that the frequency information is not affected by carrying briefcase.

6 Summary

We proposed a new method to compare gait image sequences. The characteristics of the gait are extracted from a gait volume using Fourier transform. We use the principal frequencies in the frequency domain for matching gait volumes. Thus, the data size and computational cost become quite smaller than the original gait volume. It works better than matching in the temporal domain, and the computational cost is small because the temporal alignment is not necessary.

This advantage is suitable for tracking problem. Moreover, it is shown that the spatial information is also important to discriminate gait image sequences. For future work, we analyze the effect of other factors, for example, a viewing direction and clothes.

References

1. Yam, C., Nixon, M., Carter, J.: Automated person recognition by walking and running via model-based approaches. Pattern Recognition **37** (2004) 1057–1072
2. Cunado, D., Nixon, M., Carter, J.: Automatic extraction and description of human gait models for recognition purposes. Computer Vision and Image Understanding **90** (2003) 1–41
3. Urtasun, R., Fua, P.: 3d tracking for gait characterization and recognition. In: Proc. Sixth IEEE International Conference on Automatic Face and Gesture Recognition. (2004) 17–22
4. Bobick, A., Johnson, A.: Gait recognition using static activity-specific parameters. In: Proc. Computer Vision and Pattern Recognition. (2001)
5. Lee, L., Grimson, W.: Gait analysis for recognition and classification. In: Proc. Fifth IEEE International Conference on Automatic Face and Gesture Recognition. (2002) 155–162
6. Niyogi, S., Adelson, E.: Analyzing and recognizing walking figures in xyt. In: Proc. Computer Vision and Pattern Recognition. (1994) 469–474
7. Murase, H., Sakai, R.: Moving object recognition in eigenspace representation: gait analysis and lip reading. Pattern Recognition Letters **17** (1996) 155–162
8. Little, J., Boyd, J.: Recognizing people by their gait: The shape of motion. Videre **1** (1998)
9. Liu, F., Picard, R.: Finding periodicity in space and time. In: Proc. the Sixth International Conference on Computer Vision. (1998) 376–383
10. BenAbdelkader, C., Culter, R., Nanda, H., Davis, L.: Eigengait: Motion-based recognition people using image self-similarity. In: Proc. AVBPA. (2001)
11. Liu, Y., Collins, R., Tsin, Y.: Gait sequence analysis using frieze patterns. In: Proc. the 7th European Conference on Computer Vision. Volume 2. (2002) 657–671
12. Sarkar, S., Phillips, P., Liu, Z., Vega, I., Grother, P., Bowyer, K.: The humanid gait challenge problem: Data sets, performance, and analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 162–177
13. Han, J., Bhanu, B.: Statistical feature fusion for gait-based human recognition. In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2004) 842–847
14. Ohara, Y., Sagawa, R., Echigo, T., Yagi, Y.: Gait volume: Spatio-temporal analysis of walking. In: Proc. The fifth Workshop on Omnidirectional Vision, Camera Networks and Non-classical cameras (OMNIVIS2004), Prague, Czech (2004)
15. Mituyosi, T., Yagi, Y., Yachida, M.: Real-time human feature acquisition and human tracking by omnidirectional image sensor. In: Proc. IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems. (2003) 258–263

Performance Evaluation of Object Detection and Tracking in Video

Vasant Manohar[1], Padmanabhan Soundararajan[1], Harish Raju[2],
Dmitry Goldgof[1], Rangachar Kasturi[1], and John Garofolo[3]

[1] University of South Florida, Tampa, FL
{vmanohar, psoundar, goldgof, r1k}@cse.usf.edu
[2] Advanced Interfaces Inc., State College, PA
hraju@advancedinterfaces.com
[3] National Institute of Standards and Technology, Gaithersburg, MD
john.garofolo@nist.gov

Abstract. The need for empirical evaluation metrics and algorithms is well acknowledged in the field of computer vision. The process leads to precise insights to understanding current technological capabilities and also helps in measuring progress. Hence designing good and meaningful performance measures is very critical.

In this paper, we propose two comprehensive measures, one each for detection and tracking, for video domains where an object bounding approach to ground truthing can be followed. Thorough analysis explaining the behavior of the measures for different types of detection and tracking errors are discussed. Face detection and tracking is chosen as a prototype task where such an evaluation is relevant. Results on real data comparing existing algorithms are presented and the measures are shown to be effective in capturing the accuracy of the detection/tracking systems.

1 Introduction

Recent years have seen rapid development in the state-of-the-art technologies for computer vision problems. A new approach to solving these problems is frequently proposed with high claims on its performance and robustness. Evaluation of algorithms is imperative, in order that a particular technology is not oversold. From a research point of view, well-established problems need standard databases with established benchmark performances, evaluation protocols and scoring methods available.

Object detection and tracking is a key computer vision topic, which focuses on detecting the position of a moving object in a video sequence. It is the first step accomplished by a event recognition system that extracts semantic content from video.

There have been many efforts towards empirical evaluation of object detection and tracking [1, 2, 3, 4, 5, 6, 7, 8]. These works either present a single measure that concentrates on a particular aspect of the task or a suite of measures that look at different aspects. While the former approach cannot capture the performance

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 151–161, 2006.

of the system in its entirety, the latter results in a multitude of scores which cannot be easily comprehended in assessing the performance of the system.

Similarly, while evaluating tracking systems, earlier approaches either concentrate on the spatial aspect of the task, i.e., assess correctness in terms of number of trackers and locations in frames [4, 7], or the temporal aspect which, emphasizes on maintaining consistent identity over long periods of time [2]. In the very recent works of [3, 8], a spatio-temporal approach towards evaluation of tracking systems is adopted. However, these approaches do not provide the flexibility to adapt the relative importance of each of these individual aspects. Finally, majority of these undertakings make little effort in actually comparing the performance of existing algorithms on real world applications using the proposed measures.

In this paper, we propose two comprehensive measures that capture different aspects of the detection and the tracking task in a single score. While the detection measure assumes a spatial course, a spatio-temporal concept is the backbone of the tracking measure. By adopting a thresholded approach to evaluation (See Secs 3.1 and 3.2), the relative significance of the individual aspects of the task can be modified. In the end, face detection and tracking is picked as an exemplar task for evaluation and select algorithm performances are compared on a reasonable corpus.

The remainder of the paper is organized in the following manner. Section 2 briefs the ground truth annotation process which is vital to evaluation. Section 3 describes the proposed comprehensive measures for detection and tracking. Section 4 explains the one-to-one mapping which is an integral part of this evaluation. Section 5.1 details the experimental results describing the behavior of the measures for different types of detection and tracking errors. Section 5.2 discusses and compares the results of three face detection and two face tracking algorithms on a data set containing video clips from boardroom meetings. We conclude and summarize the findings in Section 6.

2 Ground Truth Annotations

Clearly, the first step towards carrying out a scientific evaluation is to have a valid ground truth. More importantly, the approach taken towards annotation decides the evaluation technique. It has been well observed in the research community that a universal approach to annotation/evaluation cannot be adopted across domains. The main reason being the fact that features rich in a particular domain might not be discernible in a different domain.

In this paper, the method used for ground truthing is one in which objects are bounded by a geometric shape, such as rectangles, polygons or ellipses. Features of the object will be used as guides for marking the limits of the edges. If the features are occluded, which is often the case, the markings are approximated. Unique IDs are assigned to individual objects and are consistently maintained over subsequent frames. Face, text and person detection/tracking in broadcast news segments and meeting videos are few examples of the task-domain pairs where such an approach is often adopted.

There are many free and commercially available tools which can be used for ground truthing videos such as Anvil, VideoAnnex, ViPER [9], etc... In our case, we used ViPER (Video Performance Evaluation Resource), a ground truth authoring tool developed by the University of Maryland.

Fig 1 shows a sample annotation using ViPER for face in a broadcast news segment.

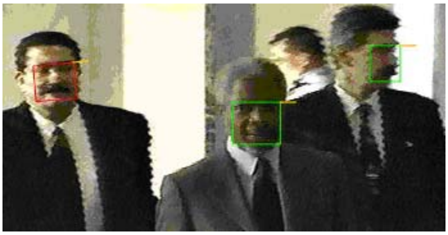


Fig. 1. Sample annotation of face in broadcast news using rectangular boxes. Facial features such as eyes and lower lip are used as guides to marking the edges of the box. Internal Data Structure maintains a unique Object ID for each of the faces shown which helps in measuring the tracking performance. Courtesy: CNN News.

A fact that has been well appreciated by the community is the need for reliable ground truth for genuine evaluations. To assure quality in the ground truth, 10% of the entire corpus was doubly annotated and checked for quality using the evaluation measures.

3 Performance Measures

The proposed performance measures are primarily area-based and depends on the spatial overlap between the ground truth and the system output objects to generate the score. In order that we get the best score of an algorithm's performance, we perform a one-to-one mapping between the ground truth and the system output objects such that the metric scores are maximized. All the measure scores are normalized such that the best performance gets a score of 1 and the worst performance gets a score of 0.

Secs 3.1 and 3.2 discuss the frame based detection measure and the sequence based tracking measure respectively, while Sec 4 briefs the one-to-one matching strategy.

The following are the notations used in the remainder of the paper,

- G_i denotes the i^{th} ground truth object and $G_i^{(t)}$ denotes the i^{th} ground truth object in t^{th} frame.
- D_i denotes the i^{th} detected object and $D_i^{(t)}$ denotes the i^{th} detected object in t^{th} frame.
- $N_G^{(t)}$ and $N_D^{(t)}$ denote the number of ground truth objects and the number of detected objects in frame t respectively.

- N_G and N_D denote the number of unique ground truth objects and the number of unique detected objects in the given sequence respectively. Uniqueness is defined by object IDs.
- N_{frames} is the number of frames in the sequence.
- N^i_{frames} is the number of frames the ground truth object (G_i) or the detected object (D_i), depending on the context, existed in the sequence.
- $N^{(t)}_{mapped}$ is the number of mapped ground truth and detected objects in frame t while N_{mapped} is the number of mapped ground truth and detected objects in the whole sequence.

3.1 Detection – Frame Based Evaluation

A good detection measure should capture the performance in terms of both overall detection (number of objects detected, missed detects and false alarms) and goodness of detection for the detected objects, i.e., spatial accuracy (how much of the ground truth is detected) and spatial fragmentation (object splits and object merges).

The Sequence Frame Detection Accuracy **(SFDA)** is a frame-level measure that penalizes for fragmentations in the spatial dimension while accounting for number of objects detected, missed detects, false alarms and spatial alignment of system output and ground truth objects. For a given frame, the Frame Detection Accuracy **(FDA)** measure calculates the spatial overlap between the ground truth and system output objects as a ratio of the spatial intersection between the two objects and the spatial union of them. The sum of all the overlaps is normalized over the average of the number of ground truth and detected objects. For a single frame t where there are $N^{(t)}_G$ ground truth objects and $N^{(t)}_D$ detected objects , we define $FDA(t)$ as,

$$FDA(t) = \frac{\text{Overlap Ratio}}{\left[\frac{N^{(t)}_G + N^{(t)}_D}{2}\right]} \tag{1}$$

$$\text{where, Overlap Ratio} = \sum_{i=1}^{N^{(t)}_{mapped}} \frac{|G^{(t)}_i \bigcap D^{(t)}_i|}{|G^{(t)}_i \bigcup D^{(t)}_i|} \tag{2}$$

Here, the $N^{(t)}_{mapped}$ is the number of mapped objects, where the mapping is done between objects which have the best spatial overlap in the given frame t.

In order to measure the detection performance for the whole sequence, the FDA is calculated over all the frames in the sequence and normalized to the number of frames in the sequence where at least a ground truth or a detected object exists. This way of normalization accounts for both missed detects and false alarms. We thus obtain the Sequence Frame Detection Accuracy (**SFDA**) which can be expressed as,

$$SFDA = \frac{\sum_{t=1}^{t=N_{frames}} FDA(t)}{\sum_{t=1}^{t=N_{frames}} \exists(N^{(t)}_G \; OR \; N^{(t)}_D)} \tag{3}$$

Fig 2 shows the effect of spatial inaccuracies (missed object region) and temporal inaccuracies (missed object frames as against object-ID mismatch which does not have any effect on the detection measure as long as the detected object spatially aligns with the ground truth.) on SFDA for a video sequence (approximately 2500 frames) containing 1 object (typically the case with close-up face videos). Here, spatial overlap ratio is defined as the ratio of the spatial intersection of the two boxes to the spatial union of them. Temporal overlap ratio is defined as the ratio of the number of frames the object was detected in to the number of frames the ground truth object existed. We can observe that given a single object, the spatial and temporal inaccuracies (missed detects at the frame level) have a linear effect on the detection measure.

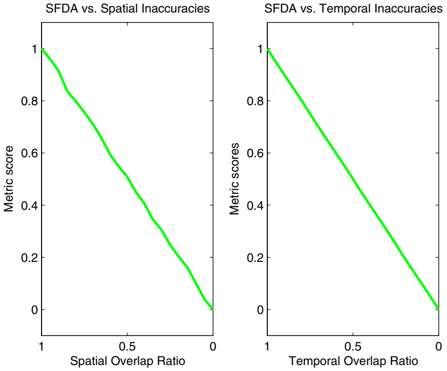


Fig. 2. Effect of spatial and temporal inaccuracies on the detection measure (SFDA) for a sequence containing a single object

Relaxing Spatial Alignment. For many systems, it would be sufficient to just detect the presence of an object in a frame, and not be concerned with the spatial accuracy of detection. To evaluate such systems, we propose a thresholded approach to evaluation of detection. Here, the detected object is given full credit even when it overlaps just a portion of the ground truth. *OLP_DET* is the spatial overlap threshold.

$$\text{Overlap Ratio Thresholded} = \sum_{i=1}^{N_{mapped}^{(t)}} \frac{Ovlp\_Thres(G_i^{(t)}, D_i^{(t)})}{|G_i^{(t)} \cup D_i^{(t)}|} \tag{4}$$

where,

$$Ovlp\_Thres(G_i^{(t)}, D_i^{(t)}) = \begin{cases} |G_i^{(t)} \cup D_i^{(t)}|, & \text{if } \frac{|G_i^{(t)} \cap D_i^{(t)}|}{|G_i^{(t)}|} \\ & \geq OLP\_DET \\ |G_i^{(t)} \cap D_i^{(t)}|, & otherwise \end{cases}$$

The threshold for a given application is derived from spatial disagreements between the annotators in the 10% double annotated data. The motivation behind this is to eliminate the error in the scores induced due to ground truth inconsistencies. Also, this way of arriving at the spatial threshold reflects the difficulties in how humans perceive the task.

3.2 Tracking – Sequence Based Evaluation

In this paper, tracking consists of simply identifying detected objects across contiguous frames. The task is similar to detection, with detected objects linked by a common identity (object IDs) across frames. Therefore, objects which leave the scene and return later in the sequence are not identified as the same object. But, occluded objects are to be treated as the same object. However, tracking is optional during occlusion. Frames in which the object is occluded are marked with special flags during annotation and these frames are excluded from evaluation.

Unlike detection, this is a spatio-temporal task and its performance can be assessed with a measure similar to the Sequence Frame Detection Accuracy measure described in Sec 3.1. The significant difference between the measures is that in detection tasks the mapping between the system output and reference annotation objects is optimized on a frame-by-frame basis, whereas for tracking, the mapping is optimized at a sequence level. One of the advantages of making this task highly parallel to the detection task is that the SFDA measure can also be applied to the tracking output to quantify the performance degradation due to mis-identification of objects across frames.

A good tracking measure should capture the performance in terms of both overall tracking (number of objects detected and tracked, missed detects and false alarms) and goodness of track for the detected objects, i.e., spatial and temporal accuracy (how much of the ground truth is detected and in how many frames) and spatial (object splits, object merges) and temporal fragmentation (discontinuous tracking).

The Sequence Track Detection Accuracy **(STDA)** is a spatio-temporal measure which penalizes fragmentations in both the temporal as well as the spatial dimensions while accounting for number of objects detected and tracked, missed objects and false alarms. A one-to-one mapping between the ground truth and the system output objects by computing the measure over all the ground truth and detected object combinations and using an optimization strategy to maximize the overall score for the sequence [see Sec 4]. The STDA is then calculated as,

$$STDA = \sum_{i=1}^{N_{mapped}} \frac{\sum_{t=1}^{N_{frames}} \left[\frac{|G_i^{(t)} \cap D_i^{(t)}|}{|G_i^{(t)} \cup D_i^{(t)}|} \right]}{N_{(G_i \cup D_i \neq \emptyset)}} \tag{5}$$

Analyzing the numerator of Eq 5, we observe that it is merely the overlap of the detected object over the ground truth, which is very similar to Eq 2. The only difference is that, in tracking we measure the overlap in the spatio-temporal dimension while in detection the overlap is in the spatial dimension alone. The value of TDA is influenced by the ability of an algorithm to detect and consistently track an object in the sequence.

The STDA is a measure of tracking over all the objects in the sequence. It can take a maximum value of N_G, which is the number of ground truth objects in the sequence. We define Average Tracking Accuracy **(ATA)**, which can be termed as the **STDA** per object, as

$$ATA = \frac{STDA}{\left[\frac{N_G + N_D}{2} \right]} \tag{6}$$

It can be readily realized that for a given object, the ATA exhibits a direct linear dependence on spatial and temporal imperfections, as was the case with the SFDA (See Fig 2).

Relaxing Detection Penalty. At times it is desirable to measure the tracking aspect of the algorithm and not be concerned with the detection accuracy. In this case, we can relax the detection penalty by using an area thresholded approach similar to Sec 3.1. In the equation described here, we introduce a threshold here namely, OLP_TRK.

$$TDA\_T(i) = \sum_{t=1}^{N_{frames}} \frac{Ovlp\_Thres(G_i^{(t)}, D_i^{(t)})}{|G_i^{(t)} \cup D_i^{(t)}|} \tag{7}$$

where,

$$Ovlp\_Thres(G_i^{(t)}, D_i^{(t)}) = \begin{cases} |G_i^{(t)} \cup D_i^{(t)}|, \text{ if } \frac{|G_i^{(t)} \cap D_i^{(t)}|}{|G_i^{(t)}|} \\ \qquad\qquad\qquad \geq OLP\_TRK \cdot \\ |G_i^{(t)} \cap D_i^{(t)}|, \, otherwise \end{cases}$$

4 Matching Strategies

From Eqs 2 and 5, it is apparent that both the detection and the tracking measures distinguish between individual objects at the frame and sequence level respectively. A valid score can be obtained only when there is a unique one-to-one mapping of ground truth and detected objects using some optimization. Potential strategies to solve this assignment problem are the weighted bi-partite graph matching and the Hungarian algorithm [10].

There are many variations of the basic Hungarian strategy most of which exploit constraints from specific problem domains. The algorithm has a series of steps which are followed iteratively and it has a polynomial time complexity. Specifically some implementations have $O(N^3)$ complexity. Faster implementations have been known to exist; the current best bound is $O(N^2logN+NM)$ [11]. In our case, the matrix to be matched is usually sparse and this fact could be taken advantage of by implementing a hash function for mapping sub-inputs from the whole set of inputs.

5 Results and Analysis

5.1 Experiments

There are many aspects of an algorithm that affect the final scores of the detection and the tracking measure. For an object detection and tracking task the errors that can affect the metric scores can be due to a single or a combination of the following errors - spatial inaccuracy, temporal inaccuracy, missed detects and false alarms. To measure the influence of all of these factors at the same time will not reflect the behavior of the measures to individual errors. Hence, in the following sections, we observe the performance of the measures by systematically handling one error at a time. We have developed an evaluation tool, which, in addition to calculating the detection and tracking scores, will also output the contribution of the above mentioned errors to the final score. This can be used for diagnostic purposes by algorithm developers to identify strengths and weaknesses of an approach and also for achieving optimal parameter settings for the algorithm.

Since we already looked at the effect of spatial and temporal inaccuracies in Fig 2, we will just investigate the effect of missed detects and false alarms in this section.

Effect of Missed Detects. In this experiment, we consider a video sequence (approximately 4500 frames) which has 75 objects that vary in their frame persistence. As against the meeting room domain where the objects persist in a longer framespan, in this case the objects stay in the scene for a short duration of time. This is typical for face, text, person and vehicle detection/tracking in broadcast news domains.

Fig 3 illustrates the performance of the measures for missed objects in the video sequence. Here, for all objects other than the missed object, we assume that they are detected and tracked ideally. Fig 3 also shows the corresponding frame persistence of the object that is missed from the ground truth. We can observe a uniform degradation of the ATA score while the SFDA score exhibits a non-uniform behavior. Clearly, the SFDA score is influenced by temporally predominant objects (existing in more frames) in the sequence, while the ATA score is independent of the frame persistence of objects. Given an ideal detection and tracking for the remaining objects in the sequence, we can analytically

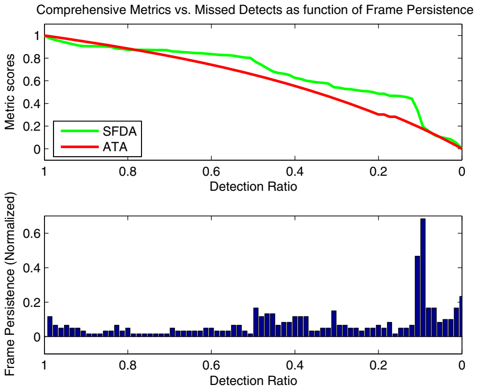


Fig. 3. Effect of missed detects on the comprehensive measures (SFDA, ATA) for a sequence containing 75 objects. The figure shows the corresponding object's frame persistence which was missed from the ground truth. For all the objects not missed, we assume ideal detection and tracking.

characterize the SFDA and the ATA measures for missed detects as shown in Eqs 8 and 9.

$$\text{SFDA} = \frac{\sum_{i=1}^{N_D} N_{frames}^i}{\frac{\sum_{i=1}^{N_D} N_{frames}^i + \sum_{j=1}^{N_G} N_{frames}^j}{2}} \tag{8}$$

$$\text{ATA} = \frac{N_D}{\left\lceil \frac{N_G + N_D}{2} \right\rceil}. \tag{9}$$

Effect of False Alarms. Having looked at the effect of missed detects on the SFDA and the ATA, it is fairly straightforward to imagine the effect of false alarms on the measure scores. Given an ideal detection and tracking for all the objects in the sequence, we can analytically characterize the SFDA and the ATA measures for false alarms as shown in Eqs 10 and 11.

$$\text{SFDA} = \frac{\sum_{j=1}^{N_G} N_{frames}^j}{\frac{\sum_{i=1}^{N_D} N_{frames}^i + \sum_{j=1}^{N_G} N_{frames}^j}{2}} \tag{10}$$

$$\text{ATA} = \frac{N_G}{\left\lceil \frac{N_G + N_D}{2} \right\rceil} \tag{11}$$

Just as missing a predominantly occurring object decreases the SFDA score by a higher extent, introducing an object in a large number of frames affects the SFDA score more. However, the ATA score is affected by the number of unique objects (different object IDs) inserted into the sequence.

5.2 Face Detection and Tracking Evaluation

In this section, we describe the test-bed that we use in our evaluation of face detection and tracking algorithms. We compared three face detection algorithms and two face tracking algorithms. The algorithm outputs were obtained from the original authors and thus can be safely assumed that the reported outputs are for the optimal parameter settings of the algorithm without any implementation errors. For anonymity purposes, these algorithms will be referred to as Algo 1, Algo 2 and Algo 3. The source video was in MPEG-2 standard in NTSC format encoded at 29.97 frames per second at 720x480 resolution.

The algorithms were trained on 50 clips, each averaging about 3 minutes (approx. 5400 frames) and tested on 20 clips, whose average length was the same as that of the training data. The ground truth was provided to algorithm developers for the 50 clips to facilitate training of algorithm parameters.

Fig 4 shows the SFDA scores of the three face detection algorithms on the 20 test clips. It also reports the SFDA scores thresholded at 10% spatial overlap, missed detects and false alarms associated with each sequence. By adopting a thresholded approach, we alleviate the effect of errors caused due to spatial anomalies. Thus, the errors in the thresholded SFDA scores are primarily due to missed detects and false alarms. One can observe a strong correlation between the SFDA scores and the missed detects/false alarms. Results show that Algo 1

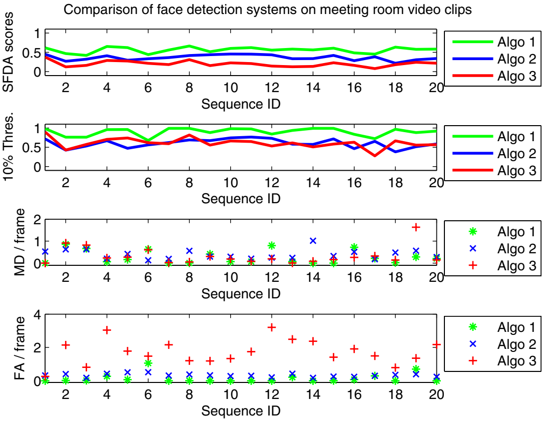


Fig. 4. Evaluation results of three face detection systems. Missed Detects (MD) and False Alarms (FA) are normalized with respect to total number of evaluation frames.

outperforms the other algorithms on all the test clips. It has good localization accuracy in addition to low missed detection and false alarms rate.

Fig 5 shows the ATA scores for the two face tracking systems on the test set. Additionally, ATA scores thresholded at 10% spatial overlap, missed detects and false alarms associated with each sequence are reported. It can be observed that, though Algo 1 has lesser identification errors and false alarm rates, there is certainly scope and promise for improvement in the performance. Results show that inconsistent identification and induction of sporadic false alarms are detrimental to performance of tracking systems.

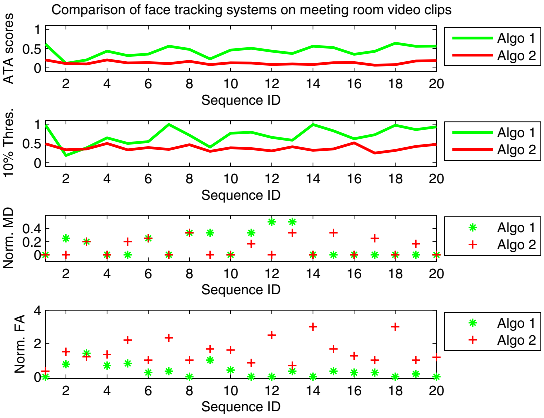


Fig. 5. Evaluation results of two face tracking algorithms. Missed Detects and False Alarms are normalized with respect to total number of unique ground truth objects in the sequence.

6 Conclusions

A comprehensive approach to evaluation of object detection and tracking algorithms is proposed for video domains where an object bounding approach to ground truth annotation is followed. An area based metric, that depends on spatial overlap between ground truth objects and system output objects to generate the score, is proposed in the case of an object bounding annotation. For the detection task, the SFDA metric captures both the detection capabilities (number of objects detected) and the goodness of detection (spatial accuracy). Similarly, for the tracking task, both the tracking capabilities (number of objects detected and tracked) and the goodness of tracking (spatial and temporal accuracy) are accounted by the ATA metric. By decomposing the performance in terms of its components, algorithm developers can analyze the robustness and shortcomings of a given approach. Evaluation results of face detection and tracking systems on meeting room video clips show the effectiveness of the metrics in capturing the performance.

References

1. Antani, S., Crandall, D., Narasimhamurthy, A., Mariano, V.Y., Kasturi, R.: Evaluation of Methods for Detection and Localization of Text in Video. In: Proceedings in International Workshop on Document Analysis Systems. (2000)
2. Black, J., Ellis, T.J., Rosin, P.: A Novel Method for Video Tracking Performance Evaluation. In: Proceedings of IEEE PETS Workshop. (2003)
3. Brown, L.M., Senior, A.W., Tian, Y., Connell, J., Hampapur, A., Shu, C., Merkl, H., Lu, M.: Performance Evaluation of Surveillance Systems Under Varying Conditions. In: Proceedings of IEEE PETS Workshop. (2005)
4. Collins, R., Zhou, X., Teh, S.: An Open Source Tracking Testbed and Evaluation Web Site. In: Proceedings of IEEE PETS Workshop. (2005)
5. Fisher, R.B.: The PETS04 Surveillance Ground-Truth Data Sets. In: Proceedings of IEEE PETS Workshop. (2004)
6. Hua, X., Wenyin, L., Zhang, H.: Automatic Performance Evaluation for Video Text Detection. In: Proc. International Conference on Document Analysis and Recognition. (2001)
7. Nascimento, J., Marques, J.: New Performance Evaluation Metrics for Object Detection Algorithms. In: Proceedings of IEEE PETS Workshop. (2004)
8. Smith, K., Gatica-Perez, D., Odobez, J., Ba, S.: Evaluating Multi-Object Tracking. In: Proceedings of IEEE Empirical Evaluation Methods in Computer Vision Workshop. (2005)
9. Doermann, D., Mihalcik, D.: Tools and Techniques for Video Performance Evaluation. In: ICPR. Volume 4. (2000) 167–170
10. Munkres, J.R.: Algorithms for the Assignment and Transportation Problems. J. SIAM **5** (1957) 32–38
11. Fredman, M.L., Tarjan, R.E.: Fibonacci Heaps and their uses in Improved Network Optimization Algorithms. Journal of ACM **34** (1987) 596–615

Vehicle Detection Using Double Slit Camera

Shunji Katahara and Masayoshi Aoki

Dept. of Computer and Information Science,
Faculty of Science and Technology, SEIKEI Univ.,
Kichijoji-kitamachi 3, Musashino-shi, Tokyo, 180-8633, Japan
{katahara, masa}@st.seikei.ac.jp

Abstract. We propose one-directional traffic flow measurement method using double slit camera. Two slit cameras are installed in overhead location with longitudinal alignment. They shoot real traffic scene in downward direction. Slit camera outputs pseudo two-dimensional image that consists of space domain and time domain. We detect vehicles from statistical pixel value of each line of a slit. Standard deviation is effective to detect bright color vehicles. We use the changes of a standard deviation and a change of an average as well as the standard deviation to detect dark color vehicles. We detect traffic flow parameters such as occupancy, time headway and time between two cars using slit camera. In double slit configuration, we detect spot speed of vehicles by the time difference of its appearance at each slit. We estimate vehicle length by normalizing the vehicle region. We also divide vehicles into type of vehicle by length.

1 Introduction

Traffic flow is counted using various vehicle detectors. Vehicle detectors are generally divided into three groups by the principle of detection. First group detectors sense the pressure when vehicle steps on a sensor, second group detectors detect a magnetic disturbance on a loop coil when vehicle passing through and third group detectors detect the reflection of a beam from vehicle. Vehicle speed is detected using Doppler radar or multiple detectors located between two points [1], [2]. Computer vision is also applied to traffic flow measurement [3], [4]. Analyzing a image sequence, a lot of traffic information such as number of vehicles, speed of vehicles, space headway are extracted in addition to multiple lanes. Area sensor is generally used as an imaging device to apply traffic flow measurement. The area sensor is suitable to detect two dimensional traffic flow such as intersection and junction.

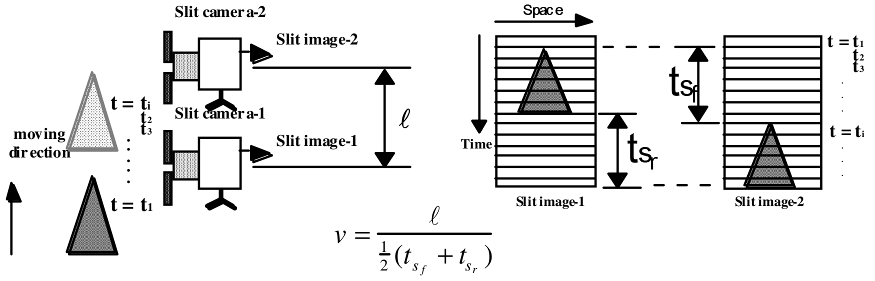
Traffic flow except intersection is regarded as one-dimensional motion, therefore, line sensor is suitable to detect one directional traffic flow [5]. Line sensor camera outputs pseudo two-dimensional image that consists of space domain and time domain. We introduce three measures to detect vehicles in a real traffic scene. The first one is a standard deviation of pixel values for detecting bright color vehicles. The second and third are the change of a standard deviation and the change of an average respectively, for detecting dark color vehicles. We also detect some traffic flow parameters from time axis information of a slit image. In double slit configuration, we detect spot speed of vehicles, and estimate vehicle length. We classify vehicle into type of vehicle by length.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 162 – 170, 2006.

2 Slit Camera

Slit camera is originally developed for sport event to record and decide goal order. One directional motion which is perpendicular to a slit will be recorded as a two dimensional image, where a slit direction corresponds to a space axis and the direction perpendicular to the slit corresponds to time axis. When the object moves faster than a film running speed at the slit, the object on the slit image comes out shrinking along time domain. When the object moves same as a film running speed, the object on the slit image comes out undistorted image. When the object moves slower than a film running speed, the object on the slit image comes out extending along time domain. The direction of motion does not affect in a slit image, therefore, the result is just the same as if both objects move opposite direction each other.

To be briefly, line sensor camera is a slit camera that the screen material is replaced a conventional photo film with CCD array. In industrial applications, it is used to record and analyze shapes of parts on a belt conveyer. It is also useful to record a very long object such as a train. In ITS world, this is a good tool for one directional traffic measurement. In those applications, the slit camera position is fixed. If we move a slit camera, we can obtain 360-degree panorama (rotation), wall or road side street scene (translation). If an on-board slit camera is looking down to the road surface and the slit is aligned to latitude direction, longitudinal vehicle motion will give road surface unfolding image.



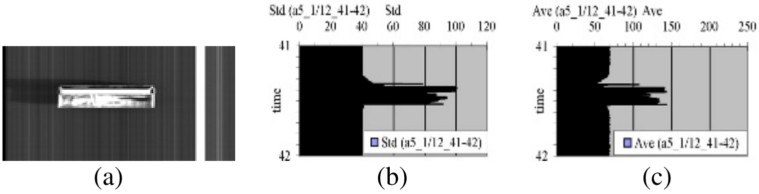
$$v = \frac{\ell}{\frac{1}{2}(t_{s_f} + t_{s_r})}$$

Fig. 1. Principle of vehicle detection using double slit

3 Vehicles in Slit Image

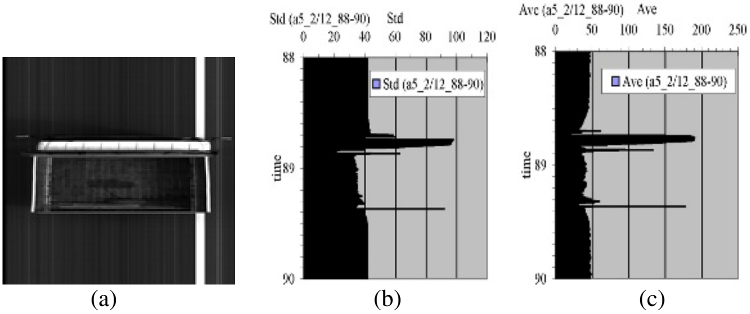
We aim to detect vehicles from each of the scan lines. We study relationship between vehicle's slit image and it's statistical pixel value of scan line, under the different finish of vehicle's body and different daylight. In Fig. 2 ~ Fig. 6, left images show various type of vehicle's image at different daylight, center figures show standard deviation of each scan line, and right figures show average of each scan line.

1) Standard deviation of road surface keeps almost constant, even if daylight slightly changes (Fig. 2 ~ 6).
2) Average of pixel value varies if daylight changes (Fig. 2 ~ 6).
3) Standard deviation becomes an effective measure for detecting bright color vehicle (Fig. 2).
4) Non-glossy object such as fabric top cover has low standard deviation (Fig. 3).
5) Standard deviations become lower, when dark color vehicle is took shot in under-exposure or relatively less daylight (Fig. 6).



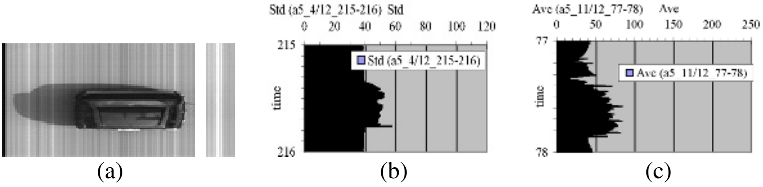
(a) (b) (c)

Fig. 2. Std. and Ave. of bright color vehicle-1



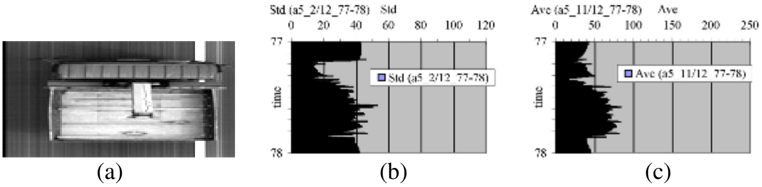
(a) (b) (c)

Fig. 3. Std. and Ave. of bright color vehicle-2



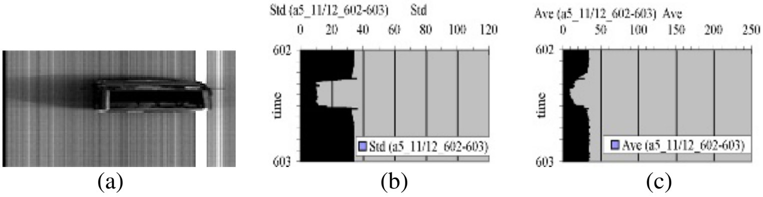
(a) (b) (c)

Fig. 4. Std. and Ave. of dark color vehicle-3



(a) (b) (c)

Fig. 5. Std. and Ave. of bright color vehicle-4 at under-exposure



(a) (b) (c)

Fig. 6. Std. and Ave. of dark color vehicle-5 at under-exposure

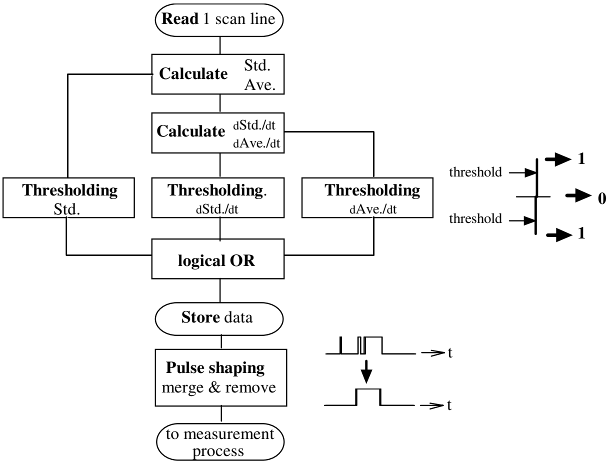
4 Outline of Method

4.1 Measures for Vehicle Detection

We use statistical pixel value of a scan line to detect vehicle. As mentioned above, standard deviation of road surface keeps constant even if daylight slightly changes. Standard deviation is efficient to detect bright color vehicles. Average of road surface depends on a daylight change, so average itself is not enough for measure. We use a change of average as a measure that is less-sensitive about daylight change. We also use a change of standard deviation to improve performance for detecting front and rear edge of vehicles.

4.2 Vehicle Detection

Fig. 7 shows flow of vehicle detection. In order to detect vehicles in each scan line, we use three measures for detection; standard deviation, change of standard deviation and change of average. We tentatively determine vehicle in a scan line when these measures are larger than or smaller than the designated thresholds. We add a flag of which 1 as including vehicle or 0 as not including vehicle, to the scan line number. These flagged line information are unified by logical OR, then merged by the duration between 1s, and replaced 1 to 0 by the isolation. After detection process, this information is send to measurement process to count traffic.



Ave(t) : average of pixel value at line t

$$\mathrm{Ave}(t) = \frac{1}{n}\sum_{x=0}^{n} I(x,t)$$

Std(t) : standard deviation of pixel value at line t

$$Std(t) = \sqrt{Std(t)^2}\,,\quad Std(t)^2 = \frac{1}{n}\sum_{x=0}^{n} I(x,t)^2 - \mathrm{Ave}(t)^2$$

n : resolution of space domain

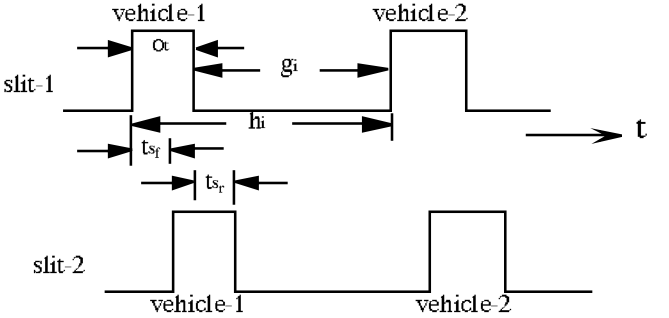
I(x,t) : pixel value

Fig. 7. Flow of vehicle detection

4.3 Traffic Flow Measurement

In Fig. 8, pulse duration corresponds to occupancy, pulse spacing corresponds to time between two cars and interval between a leading edge of a pulse and next leading edge of a pulse corresponds to time headway. Time difference of a leading edge of a pulse (front edge of vehicles) or a trailing edge of a pulse (rear edge of vehicles) between the slits corresponds to a time required passing through the slits. We can get occupancy; **Ot**[sec], time between two cars; **gi**[sec], time headway; **hi**[sec] and time difference of its appearance at each slit; **ts**[sec]. Spot speed; **vt** is calculated by **vt=ds/ts**[m/sec], because distance between slits; **ds**[m] is already known.

An equivalent film running speed of the camera; **vslit** is derived from line capturing speed; **cslit**[line/sec] (or scanning time of a line) and resolution of the camera; **rt**[m/line], and as follows **vslit=rt·cslit**[m/sec]. Therefore modification coefficient for restoring non-extended and non-contracted slit image of vehicles; **k** becomes **k=vt/vslit**. Vehicle length; ℓ is estimated by **ℓ=ot·rt·k**[m] using occupancy, resolution and modification coefficient. We also obtain traffic volume; **Q**, rate of flow; **q**[/h], occupancy: **Qt**[%], time mean speed; $\overline{v_t}$ [m/sec], average time headway; $\overline{h}$ [sec], average time between two cars; $\overline{g}$ [sec], as macroscopic traffic flow parameters for designated period.



Ot: occupancy
gi: time between two cars
hi: time headway
ts: time difference of its appearance at each slit
(**tsf:** leading edge, **tsr:** trailing edge)

Fig. 8. Parameters derived from wave forms

5 Experiment

5.1 Experimental Set-Up

We install two line sensor cameras on a pedestrian overpass at 6.1 meters height from road surface. Both cameras look down to the road surface, and the slits are aligned to latitude direction with 2.6 meters distance between slits. The line sensor camera equipped 10.24[mm]width by 10[mm]length CCD array and 17[mm] focal length optical lens, so range of vision becomes 3.7[m]width by 3.6[mm]length in each line, and resolution of CCD becomes 3.6[mm/pixel]. We fix on a line capturing speed of the camera; 500 [line/sec], or one line scanning time; 2 [ms].

5.2 Measures for Vehicle Detection

In order to adapt detection to some different road surface materials, we study the value about standard deviation and average of pixel value at measurement position, then decide the thresholds.

1) A scan line includes vehicle when Std. becomes less than 30 or larger than 45.
2) A scan line includes vehicle when change of Std. becomes less than -3 or larger than 3.
3) A scan line includes vehicle when change of Ave. becomes less than -6 or larger than 6.

5.3 Vehicle Detection

We tentatively detect whether a scan line includes or not includes vehicle from three measures. We add a flag of which 1 as including vehicle or 0 as not including vehicle to the scan line. The flagged line are unified by logical OR, then merged by the duration between 1s and replace 1 to 0 by the isolation. Almost vehicles keep more than 0.5 second distance against preceding vehicle, therefore candidates of vehicle within 0.5 second interval are merged as a same vehicle.

Fig. 9 show one minute slit image during 12 minutes shooting and typical example of detection process (whole slit image has 1024 pixel along space axis and 360,000 line along time axis). Fig. 9 (a) shows original slit image, (b) shows histogram equalized slit image for easy to see, (c) shows Std. of scan line, (d) shows change of Std, (e) shows Ave. of scan line, (f) shows change of Ave. (g) shows vehicle detection by three measures, logical OR and merged result respectively.

From human observation, 170 vehicles pass through the upper side camera, and 166 vehicles pass through the lower side camera during 12 minutes. We detect 165 vehicles at upper side, and 163 vehicles at lower side. Correct detection rate becomes 96 % taking into account of redundancy and insufficiency of detection results.

5.4 Traffic Flow Measurement

5.4.1 Correspondence of Vehicles Between Slits

In order to detect vehicle speed, we study correspondence of vehicles between slits. We extract and regard as corresponding vehicle that appears within one second from upper slit to lower slit (faster than 7.8 km/h between slits). We can adapt slow traffic by this interval to longer. In the speed detection stage, we can finally detect 153 vehicles, and detection rate becomes 90 %.

5.4.2 Traffic Flow Measurement

Table 1 represents typical example of occupancy, vehicle speed, vehicle length of detected vehicles with human observation results. Detected speed of vehicles includes

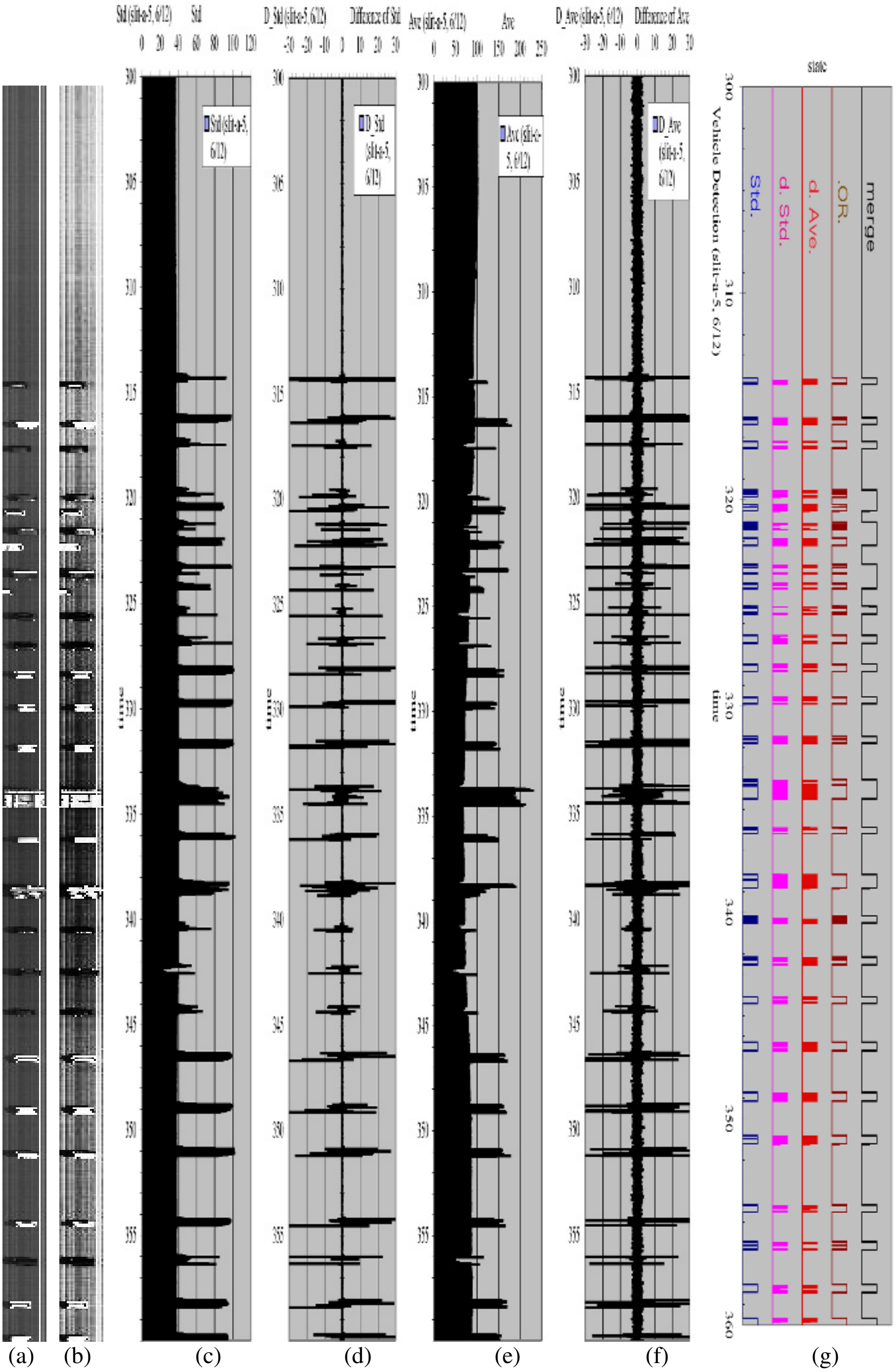


Fig. 9. Typical detection result of vehicles (5~6 minute or 30,000 line data during 12 minutes measurement)

an error of maximum 30km/h. Table 2 represents some traffic flow parameters derived from the measurement. We estimate vehicle length from occupancy and vehicle velocity, then classify vehicles into type of vehicle by length. We regard vehicles less than 2.5 [m] length as a motor cycle, less than 3.4 [m] length as a sub-compact car, less than 4.7 [m] length as a compact car, and longer than 4.7 [m] as a regular car. We classify the objects less than 1 [m] or longer than 12 [m] into irregular. Classified results show estimated lengths of many vehicles are longer than the manually detected results, due to the effect of shadow.

Table 1. Example of vehicle detection result

Vehicle No.	Correspond Vehicle		Occupancy [sec]							Vehicle speed [km/h]								Mod. Coef_cient		Vehicle length	
Slit-1	No. at Slit-2		Detected			Human observatiom				Detected			Human observation			Error		Detected	Observed	Detected	Observed
	Front edge	Rear edge	slit-1	slit-2	Average	Slit-1	slit-2	Average	error [%]	Front edge	Rear edge	Average	Front edge	Rear edge	Average	[km/h]	[%]			[m]	[m]
1	1	1	0.362	0.368	0.365	0.374	0.362	0.368	-0.8	46.8	45.2	46.0	42.3	45.2	43.7	2.3	5.2	6.2	5.9	4.8	4.6
2	2	2	0.284	0.296	0.290	0.278	0.274	0.276	5.1	60.8	55.5	58.1	55.5	57.2	56.4	1.8	3.2	7.9	7.6	4.8	4.4
3	3	3	0.238	0.248	0.243	0.234	0.226	0.230	5.7	67.0	61.7	64.4	60.8	64.8	62.8	1.6	2.5	8.7	8.5	4.5	4.1
4	4	4	0.210	0.216	0.213	0.198	0.186	0.192	10.9	58.9	56.3	57.6	52.5	57.2	54.9	2.8	5.0	7.8	7.4	3.5	3.0
5	5	5	0.374	0.388	0.381	0.350	0.346	0.348	9.5	47.4	43.7	45.6	43.2	44.2	43.7	1.9	4.3	6.2	5.9	4.9	4.3
6	6	6	0.290	0.282	0.286	0.318	0.318	0.318	-10.1	54.8	58.0	56.4	57.2	57.2	57.2	-0.8	-1.4	7.6	7.7	4.6	5.2
7	7	7	0.210	0.212	0.211	0.214	0.214	0.214	-1.4	54.8	54.0	54.4	55.5	55.5	55.5	-1.2	-2.1	7.4	7.5	3.3	3.4
8	8	8	0.276	0.276	0.276	0.278	0.270	0.274	0.7	56.3	56.3	56.3	55.5	58.9	57.2	-0.9	-1.5	7.6	7.8	4.4	4.5
10	10	10	0.218	0.340	0.279	0.350	0.338	0.344	-18.9	92.6	37.7	65.2	36.7	38.9	37.8	27.4	72.5	8.8	5.1	5.2	3.7
11	11	11	0.392	0.394	0.393	0.398	0.390	0.394	-0.3	39.7	39.3	39.5	38.9	40.5	39.7	-0.2	-0.5	5.3	5.4	4.4	4.4
12	12	12	0.306	0.300	0.303	0.370	0.358	0.364	-16.8	40.5	41.8	41.2	40.5	43.2	41.9	-0.7	-1.7	5.6	5.7	3.5	4.3
13	13	13	0.382	0.386	0.384	0.382	0.382	0.382	0.5	40.5	39.7	40.1	40.5	40.5	40.5	-0.4	-1.0	5.4	5.5	4.4	4.4
14	14	14	0.278	0.278	0.278	0.282	0.274	0.278	0.0	40.9	40.9	40.9	40.5	42.3	41.4	-0.5	-1.1	5.5	5.6	3.2	3.3
15	15	15	0.680	1.070	0.875	0.678	0.670	0.674	29.8	41.8	13.5	27.7	41.4	43.2	42.3	-14.6	-34.6	3.7	5.7	6.9	8.1
16	16	16	0.948	0.930	0.939	0.950	0.926	0.938	0.1	38.9	42.7	40.8	38.9	44.2	41.5	-0.7	-1.8	5.5	5.6	10.9	11.1
17	17	17	0.314	0.314	0.314	0.318	0.314	0.316	-0.6	43.7	43.7	43.7	45.2	46.3	45.7	-2.1	-4.5	5.9	6.2	3.9	4.1
18	18	18	0.454	0.466	0.460	0.458	0.454	0.456	0.9	47.4	44.2	45.8	42.3	43.2	42.7	3.1	7.2	6.2	5.8	6.0	5.5
19	19	19	0.382	0.382	0.382	0.386	0.382	0.384	-0.5	42.7	42.7	42.7	42.3	43.2	42.7	0.0	0.0	5.8	5.8	4.6	4.7
20	20	20	0.370	0.382	0.376	0.342	0.338	0.340	10.6	49.8	46.3	48.1	45.2	46.3	45.7	2.3	5.1	6.5	6.2	5.1	4.4
21	21	21	0.386	0.412	0.399	0.382	0.378	0.380	5.0	49.2	42.3	45.7	42.3	43.2	42.7	3.0	7.0	6.2	5.8	5.2	4.6
23	23	23	0.410	0.436	0.423	0.378	0.378	0.378	11.9	42.3	37.0	39.6	38.1	38.1	38.1	1.5	4.0	5.4	5.2	4.8	4.1
24	24	24	0.434	0.456	0.445	0.398	0.382	0.390	14.1	42.3	37.7	40.0	36.0	38.9	37.4	2.6	6.8	5.4	5.1	5.1	4.2
25	25	25	0.316	0.328	0.322	0.310	0.306	0.308	4.5	54.8	50.5	52.6	51.2	52.5	51.8	0.8	1.5	7.1	7.0	4.8	4.5
26	26	26	0.314	0.322	0.318	0.290	0.286	0.288	10.4	51.8	49.2	50.5	48.6	49.8	49.2	1.3	2.7	6.8	6.7	4.6	4.0
27	27	27	0.378	0.386	0.382	0.358	0.346	0.352	8.5	48.6	46.3	47.4	45.2	48.6	46.9	0.5	1.1	6.4	6.4	5.2	4.7
28	28	28	0.332	0.338	0.335	0.306	0.302	0.304	10.2	50.5	48.6	49.5	49.8	51.2	50.5	-1.0	-1.9	6.7	6.8	4.7	4.4
29	29	29	0.356	0.802	0.579	0.334	0.330	0.332	74.4	50.5	13.0	31.7	47.4	48.6	48.0	-16.3	-33.9	4.3	6.5	5.2	4.5
30	30	30	0.788	0.782	0.785	0.734	0.726	0.730	7.5	42.3	43.7	43.0	42.3	44.2	43.2	-0.2	-0.6	5.8	5.9	9.6	9.0

Table 2. Traffic flow measurement result and classified type of vehicles by length

	Traffic volume	Rate of traffic flow [/h]	Time occupancy	Aveeage time [sec] between two cars	Average time headway [sec]
Manual	168	840	0.074	3.90	4.20
Experiment	153	765	0.084	4.33	4.73

Average time	**Vehicle speed** [km/h]		
headway [sec]	Average	Max. speed	Min. speed
4.20	46.2	108.0	29.7
4.73	47.5	108.2	27.0

shorter than 1 m		**Type of vehicles**			longer than 12 m
Irregular	Bike	Sub-compact	Compact	Standard	Irregular
0	4	19	110	20	0
2	4	8	77	56	6

6 Conclusion

We propose a traffic flow measurement method using double slit camera. We detect vehicles at each scan line from three measures. We also count vehicles, occupancy, time headway and time between two cars. We detect vehicle speed and estimate vehicle length by double slit configuration. We classify vehicles into type of vehicle by length.

In order to improve detection, we have to prepare fine exposure slit image and consider countermeasures against effect of shadow from the vehicles. We also have to add a measure for detection using likelihood of vehicle from consistency and continuity of a line profile.

References

1. Lawrence A. Klein: Vehicle Detector Technologies for Traffic Management Applications Part 1 & Part 2, ITS Online, The Independent Forum for Intelligent Transportation Systems (1977) Available: http://www.itsonline.com/detect_1.html&detect_2.html
2. Dan Middleton, Deepak Gopalakrishna, and Mala Raman: Advances in Traffic Data Collection and Management, White Paper, BAT-02-006, Traffic Data Quality Workshop, Federal Highway Administration, January 31 (2003) Available: http://www.itsdocs.fhwa.dot.gov// JPDOCS/REPTS_TE/13766.html
3. M. Takatoo, T. Kitamura, Y Okuyama, Y. Kobayashi, K. Kikuchi, H. Nakanishi and T. Shibata: Traffic flow measurement system using image processing, Proc. SPIE Int. Soc Opt Eng, Vol. 1197, pp. 172-180 (1990)
4. N. Hashimoto, Y. Kumagai, K. Sakai, K. Sugimoto, Y. Ito, K. Sawai and K. Nishiyama: Development of an image-processing traffic flow measurement system, Sumitomo Electric Technical Review, No. 25, pp. 133-138 (1986)
5. Shunji Katahara, Tetsuro Izumi, Shota Kawamata and Masayoshi Aoki: Traffic Flow Measurement Using Double Slit Image, 9th World Congress on ITS, 3071, TP029 (2002)

Automatic Vehicle Detection Using Statistical Approach

Chi-Chen Raxle Wang and Jenn-Jier James Lien

Robotics Laboratory, Dept. of Computer Science and Information Engineering,
National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan, Taiwan
{raxle, jjlien}@csie.ncku.edu.tw
http://robotics.csie.ncku.edu.tw

Abstract. This study develops a statistical approach to the automatic detection of vehicles. Compared to traditional approaches, which consider the entire 2-dimensional vehicle region, this study uses three meaningful local features for each vehicle to perform vehicle detection. The proposed approach has a superior tolerance toward wider viewing angles and partial occlusions. Four possible models for vehicle detection are evaluated in the current training and testing processes. For the process of the best model, each local subregion projects into corresponding eigenspace and residual independent basis space with subregion position information. We further simplify the procedure steps of computing the independent component analysis (ICA) in residual space without constructing residual images in order to reduce the computational time. Then the joint probability of projection weight vectors and coefficient vectors of local subregions and positions of local subregions, is used to model the vehicle. Finally, we introduce vector quantization with a new classification method to accelerate the posterior probability calculation.

1 Introduction

Automatic license plate identification tools are invaluable for applications such as parking lot access control, easy-pass toll collection, stolen vehicle recovery, etc. Vehicle detection is an essential and integral part of vehicle plate identification. In [4], stereo is used to detect moving targets. In this approach, the target vehicle is identifiable if its grayvalues and the edges of the target region exhibit left-right symmetry. However, this method suffers when partial occlusion occurs since this results in unsymmetrical regions. Posterior probability [5] can be applied to detect moving vehicles based on their edge information. However, using edge information alone renders the vehicle detection process liable to noise and illumination effects. Furthermore, the success of the posterior probability approach relies strongly on the probability of the vehicle appearance falling within a limited range during the training process. For real-time vehicle detection applications, background subtraction method is used in the initial segmentation process of the foreground moving objects and background scenes [3]. Subsequently, vehicle templates and edge information are applied to carry out vehicle detection.

Some detection methods capture the global feature information associated with vehicle images, while others simply capture the local feature information. Although global feature approaches such as [9] and [10] perform reasonably well, in real life, individuals are able to recognize a vehicle from its local features without needing to

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 171 – 182, 2006.

see the entire vehicle. The studies of [7] and [12] have demonstrated that local feature approaches yield better detection results than their global feature counterparts. Local feature approaches such as those proposed in [13], [14], [15], and [16] and part-based approaches, e.g. [1], [6], and [8], have been successfully implemented for object, face, and vehicle detection systems. The latter approaches incorporate an interest-points detector to locate a specified region and to describe their corresponding position in the vehicle image.

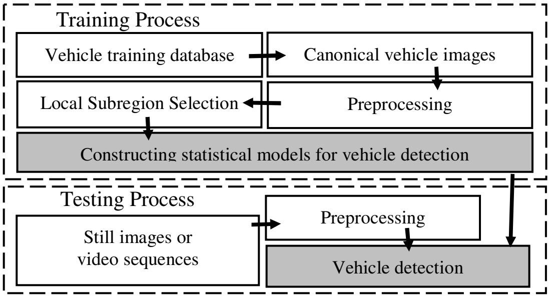
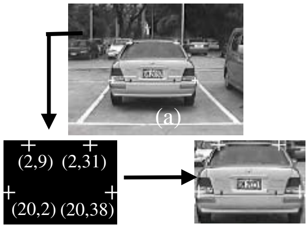

Training Process
Vehicle training database
Canonical vehicle images
Local Subregion Selection
Preprocessing
Constructing statistical models for vehicle detection
Testing Process
Still images or video sequences
Preprocessing
Vehicle detection

Fig. 1. Workflow diagram of the proposed vehicle detection system



(b) Vehicle template region: N_r*N_c pixels

(c) Canonical vehicle image

Fig. 2. Geometrical normalization and canonical vehicle image creation

Posterior probability estimation, including both vehicle and non-vehicle information, is used in [14] for the robust detection of vehicles, faces and objects of different sizes and poses. In this approach, the likelihood evaluation operation is based on the estimated joint probability of wavelet coefficients and their corresponding positions within a given region. The same authors used a similar approach in detecting vehicles and faces of different sizes and poses [15]. Their study adopted principal component analysis (PCA) and considered the position information of a 16*16-pixel subregion in the joint probability estimation procedure. Unfortunately the detection process involved in this kind of studies were rather time-consuming.

To overcome the weaknesses of the methods reported in the literature, this study utilizes posterior probability with both vehicle and non-vehicle information to conduct automatic vehicle detection. The joint probability for the maximum-likelihood estimation procedure considers both meaningful local features and their corresponding positions. This study combines the PCA space and the ICA in residual space to model the vehicle. The performance of the proposed approach is verified through a series of experimental studies. Moreover, in order to reduce the computational time required for the detection process, a vector quantization method with a new classification approach is applied to classify the training vehicle and non-vehicle images into several clusters. However, we accelerate the detection process but it won't decrease the system performance.

2 Vehicle Detection System

The vehicle detection system consists of a training process and a testing process, as shown in Figure 1. Current study considers the case of a surveillance system such as that used for access control in schools, which detects only the rear and frontal views of passing small vehicles, such as saloon and SUV. Therefore, the case of detecting passing vehicles from side perspectives, etc. is specifically excluded from the current scope. The vehicle template and canonical vehicle images in the present study are *Nr*(=32) * *Nc*(=41) (rows*columns) pixels.

To create a canonical rear-viewed vehicle image, four corner points are manually selected on the vehicle in the original training image, as shown in Figure 2.a. The vehicle image is then normalized and cropped by using the vehicle template shown in Figure 2.b and applying a process of affine transformation to the four corner points. The resulting canonical vehicle image is shown in Figure 2.c. The same procedure is adopted to generate the canonical frontal-viewed vehicle image.

2.1 Local Subregion Selection

The present system considers only local features rather than the entire vehicle in the detection process since in real life, individuals are easily able to recognize a vehicle from its local features, i.e. they do not need to see the vehicle in its entirety. Furthermore, this approach can reduce the alignment error by accommodating geometric distortions of the vehicle appearance (texture or grayvalue) to a certain extent [15]. The proposed approach also increases the detection tolerance in the event of unbalanced targets caused by uneven road surfaces or unstable input sources such as handheld video cameras. Finally, considering a local subregion can improve the overall system performance by reducing the computation of time.

Generally, the significant local features in the rear- and frontal-viewed vehicle images contain high texture components such as roofs, windshields, tail-lights (or head-lights), license plates, rear-viewed mirrors, and the wheels [12]. These features exhibit high variances in the spatial domain. However, these subregions may not always be visible in the vehicle image. For example, the rear-viewed mirrors and wheels may disappear in some situations. Moreover, the subregions around the license plate and the windshield areas are sensitive to different locations and illumination, respectively. Therefore, as shown in Figure 3, the current study opts specifically to ignore these particular significant features, and chooses instead the subregions around the roof and the tail-lights (or head-lights).

2.2 Vehicle Detection Using Posterior Probability Function

This study detects both rear- and frontal-viewed vehicles from an input image, *I*, by shifting a window, I_T, measuring N_r*N_c pixels, pixel by pixel over the entire image. The vehicle is detected if it is found within the window.

The following posterior probability function is used in the vehicle detection procedure:

$$P(vehicle \mid I_T) = \frac{P(I_T \mid vehicle)P(vehicle)}{P(I_T \mid vehicle)P(vehicle) + P(I_T \mid non-vehicle)P(non-vehicle)} \tag{1}$$

It is assumed that the prior probability is uniformly distributed, i.e. *P*(*vehicle*) = *P*(*non-vehicle*) = 0.5. It is also assumed that the likelihood probabilities *P*(I_T|*vehicle*) and *P*(I_T|*non-vehicle*) conform to a multivariate Gaussian distribution, i.e.

$$P(I_T \mid C) = \frac{\exp\left[-\frac{1}{2}(I_T - \bar{I}_{C,T})^T \Sigma^{-1} (I_T - \bar{I}_{C,T})\right]}{(2\pi)^{N/2}\left|\Sigma\right|^{1/2}} \tag{2}$$

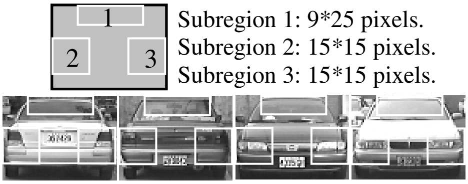

1
2
3
Subregion 1: 9*25 pixels.
Subregion 2: 15*15 pixels.
Subregion 3: 15*15 pixels.

Fig. 3. There are three local feature subregions for each canonical vehicle image

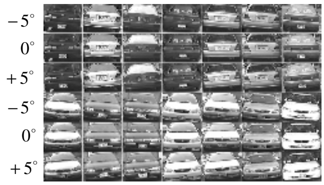


Fig. 4. Original (0^0) and synthetic canonical vehicle images (-5^0 and $+5^0$)

where C is either the vehicle class or non-vehicle class; $\bar{I}_{C,T}$ and Σ are the mean vector and the covariance matrix of all canonical training image vectors for class C, respectively; and N is the total number of vector dimensions. From equation (2), the Mahalanobis distance $d(I_T)$ [13] is given by:

$$d(I_T) = \tilde{I}_T^{\,T} \Sigma^{-1} \tilde{I}_T = \tilde{I}_T^{\,T} \left[UW^{-1}U^T \right] \tilde{I}_T \approx \tilde{I}_T^{\,T} \left[U_k W_k^{-1} U_k^T \right] \tilde{I}_T = y_k^T W_k^{-1} y_k \tag{3}$$

where $\tilde{I}_T = I_T - \bar{I}_{C,T}$, $y_k = U_k^T \tilde{I}_T$, and W_k and U_k are the first k principal components of the eigenvalue matrix W and its corresponding eigenvector matrix, U, of the covariance matrix Σ, respectively. The input window, I_T, is projected into the eigenspace U_k to generate a weight vector y_k. Therefore, the Mahalanobis distance can be represented as:

$$d(I_T) = \sum_{i=1}^{k} \frac{y_i^2}{\lambda_i} \tag{4}$$

Hence, the likelihood probability in equation (2) becomes:

$$P(I_T \mid C) = \frac{Exp\left[-\frac{1}{2} \sum_{i=1}^{k} \frac{y_i^2}{\lambda_i} \right]}{(2\pi)^{k/2} \prod_{i=1}^{k} \lambda_i^{1/2}} \tag{5}$$

2.3 Different Detection Models in the Training and Testing Processes

A separated detection process is employed for rear- and frontal-viewed vehicles. The current training database includes 275 canonical rear-viewed vehicle images and 262 frontal-viewed vehicle images. In order to develop the capability of detecting vehicles moving on uneven roads or shot by handheld video cameras, two additional in-plane roll-rotation image views, i.e. (-5^0) and ($+5^0$), are generated synthetically from the original canonical vehicle images (0^0).

Therefore, as shown in Figure 4, each canonical vehicle in the training database actually has three associated images. Furthermore, three subregions are defined within each image. In other words, the training database contains a total of 2475 canonical rear-viewed vehicle images (275 * 3(rotations) * 3(subregions)), and a total of 2358 canonical frontal-viewed vehicle images (262 * 3(rotations) * 3(subregions). The images in the database are preprocessed by affine lighting correction and histogram equalization [7]. The intensity over the entire canonical image is then normalized to zero mean and unit variance [15].

Previous studies [7], [11], [14], and [15] have shown that the use of subregions in face detection or recognition applications yields excellent results. Hence, the detection system proposed in this study operates on the basis of three independent subregions rather than over the entire vehicle region. Hence, the likelihood probability is given by:

$$P(I_T \mid C) = \prod_{i=1}^{3} P(subregion_i \mid C) \tag{6}$$

Unfortunately, this approach is computationally expensive when applying posterior probability since it involves a very high-dimensional image vector. Therefore, this study evaluates the performance of four different detection models in the current training and testing processes. Previous studies have confirmed that the PCA method employed in this study has excellent properties. First, the correlation of the neighborhood pixels remains high. Second, a larger eigenvalue implies more significant variance among the original unbiased image vectors. Third, each original image vector can be reconstructed by the linear combination of the major eigenvectors without losing significant characteristics. Furthermore, the ICA applied to the residual subregion spaces in this study also has excellent characteristics [2] and [11]. First, the ICA can capture high-order statistical information. Second, it is suitable for the modeling of non-Gaussian distributed data sets, such as those associated with the residual subregion spaces in the present study. Third, the ICA applied in the residual spaces is robust to illumination and pose variations. The following sections describe in detail the application of the four proposed detection models to rear-viewed vehicle images. However, it is noted that these models are equally applicable to the detection of frontal-viewed vehicle images.

2.3.1 1st Model: All Subregions Are Projected into One Single Eigenspace Without Position Information

In the training process, one eigenspace is generated from the 2475 subregions of the total set of canonical rear-viewed vehicle images. The first 32 principal components are captured since the accumulated eigenvalue percentage curve has a turning point at *k*=32. All of the canonical vehicle or non-vehicle subregions are then projected into this eigenspace, which consists of 32 major eigenvectors and hence reduces its dimensions from 225 to 32. Equation (6) becomes:

$$\prod_{i=1}^{3} P(subregion_i \mid C) = \prod_{i=1}^{3} P(projection_i \mid C) \tag{7}$$

where $projection_i$ represents a 32-dimensional weight vector for subregion *i*. Figure 5 presents the eigenvectors of this eigenspace. It can be seen that the 1st, 4th and 6th eigenvectors fall mainly within subregion 1, while the 2nd、3rd and 5th eigenvectors fall inside subregions 2 or 3. Finally, the posterior probability equation for the 1st model is given by:

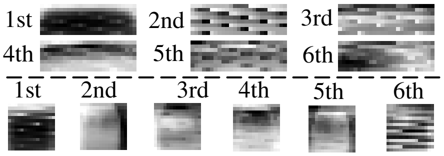


Fig. 5. The first six eigenvectors of the 1st model

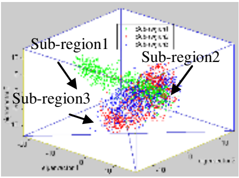


Fig. 6. Weight-vector distributions corresponding to canonical vehicle subregions in the 2nd model

$$P(vehicle \mid I_T) = \frac{\prod_{i=1}^{3} P(projection_i \mid vehicle)}{\prod_{i=1}^{3} P(projection_i \mid vehicle) + \prod_{i=1}^{3} P(projection_i \mid non-vehicle)} \tag{8}$$

In the testing process, any input window ($N_r{*}N_c$ pixels), I_T, consists of three subregions. Each subregion is projected into the eigenspace to generate a corresponding weight vector. Each of the three input subregions is then compared with the total set of canonical training subregions. Based on equations (5) and (8), it is possible to detect the existence of a vehicle inside this input window by means of the following criterion:

$$\begin{array}{ll} if & (P(vehicle \mid I_T) \geq threshold) \\ then & (vehicle\ \ exists) \\ otherwise & (vehicle\ \ doesn't\ exist) \end{array} \tag{9}$$

2.3.2 2nd Model: All Subregions Are Projected into One Single Eigenspace with Position Information

In the training process, this model uses the same eigenspace as that described in the 1st model, above. However, this model also takes into account the feature position of each subregion. The complete set of canonical vehicle subregions are classified in accordance with their positions into three separated groups of weight vectors in a 32-dimensional eigenspace. As can be seen in Figure 6, the distributions of subregions 2 and 3 are overlapped since they are symmetric in the canonical vehicle images. However, the distribution of subregion 1 is very different as a result of the apparent texture differences between itself and subregions 2 and 3. The same classification process is also applied to each of three canonical non-vehicle subregions. Taking the additional position information into account, equation (6) becomes:

$$\prod_{i=1}^{3} P(subregion_i \mid C) = \prod_{i=1}^{3} P(projection_i, pos_i \mid C) \tag{10}$$

where pos_i is the position of subregion i of the given vehicle template region. The posterior probability equation for the 2nd model becomes:

$$P(vehicle \mid I_T) = \frac{\prod_{i=1}^{3} P(projection_i, pos_i \mid vehicle)}{\prod_{i=1}^{3} P(projection_i, pos_i \mid vehicle) + \prod_{i=1}^{3} P(projection_i, pos_i \mid non-vehicle)} \tag{11}$$

In the testing process, each subregion of the input window, I_T, is compared with the canonical training subregions located at the corresponding position. The posterior probability of the input window can be evaluated from equations (5) and (11). The existence of vehicles can then be determined by assigning different threshold values in equation (9).

2.3.3 3rd Model: Each Subregion Is Projected into Corresponding Eigenspace with Position Information

In addition to taking into account the position of the subregions, this model also generates an eigenspace for each group of canonical vehicle subregions. Hence, three eigenspaces exist for the three subregion groups with different positions. For reasons of consistency, each eigenspace has 32 major eigenvectors, i.e. as in the two models presented above. Figure 7 shows the first three eigenvectors for each subregion eigenspace. Finally, each canonical vehicle or non-vehicle subregion is projected into the corresponding eigenspace to generate a weight vector. Therefore, equation (6) becomes:

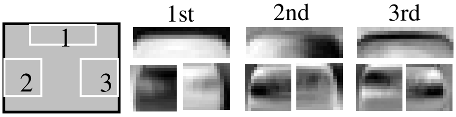


Fig. 7. The first three eigenvectors for each subregion eigenspace in the 3rd model

$$\prod_{i=1}^{3} P(subregion_i \mid C) = \prod_{i=1}^{3} P\left(projection_i^i, pos_i \mid C\right) \tag{12}$$

where $projection_i^i$ is the weight vector of the subregion *i* projected into the corresponding eigenspace *i*. The posterior probability equation for the 3rd model becomes:

$$P(vehicle \mid I_T) = \frac{\prod_{i=1}^{3} P(projection_i^i, pos_i \mid vehicle)}{\prod_{i=1}^{3} P(projection_i^i, pos_i \mid vehicle) + \prod_{i=1}^{3} P(projection_i^i, pos_i \mid non-vehicle)} \tag{13}$$

In the testing process, each subregion of the input vehicle template window, I_T, is projected into the corresponding eigenspace to generate a 32-dimensional weight vector. Hence, three weight vectors exist for each input window. The posterior probability of the input window, I_T, is calculated from equations (5) and (13). The existence of vehicles can then be determined by assigning different threshold values in equation (9).

2.3.4 4th Model: Each Subregion Is Projected into Corresponding Eigenspace and Residual Independent Basis Space with Position Information

This model applies the ICA in the residual spaces to detect the vehicle. The similar work in face recognition [11] performs well in its result. The authors construct ICA in residual space after computing the residual images by subtracting the reconstructed images from the original images. We further derive equations that simplify the procedure steps of computing the ICA in residual space without constructing residual images, and then apply Bayesian theory to detect vehicles. The equations we developed require less complicated calculations.

The independent components, which form non-orthogonal axes, describe the residual subregion spaces of the three subregion groups with different positions. The residual subregion spaces (see Figure 8.c) represent the difference between the original subregion images (see Figure 8.a) and the PCA reconstructed subregion images (see Figure 8.b). It is found that the PCA reconstructed subregions are similar to low-pass filtered versions. The residual subregion images, which contain high frequency components, are less sensitive to illumination variations.

ICA is applied in the residual subregion spaces since these spaces are non-Gaussian distributions. Therefore, to achieve a detection operation, which is robust to illumination and pose variation effects, each residual subregion image is represented by a linear combination of independent components.

Each residual subregion image, $\triangle subregion$, can be obtained by equation 14:

$$\Delta subregion = subregion - subregion' \tag{14}$$

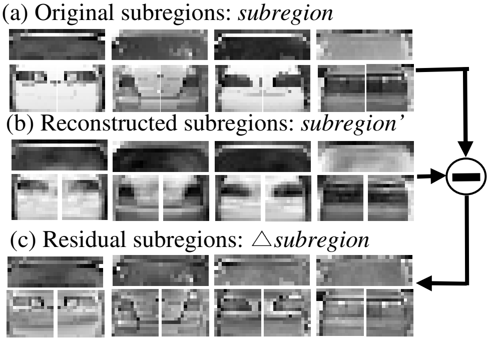
where *subregion* is the original subregion image, and *subregion*' is the PCA reconstructed subregion image. They are given by:

$$subregion = UU^T * subregion = \begin{bmatrix} U_{k'} & U_h \end{bmatrix} \begin{bmatrix} U_{k'}^T \\ U_h^T \end{bmatrix} * sugregion \tag{15.a}$$

and

$$subregion' = U_{k'}U_{k'}^{T} * subregion \qquad (15.b)$$

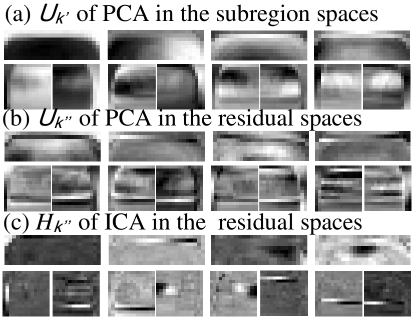
where $U_{k'}$ (see Figure 9.a) is the first k' principal components in eigenvector matrix U, U_h is the h residual principal components, and N is $k' + h$. The first k' (k'=7) components are chosen based on the Gaussian axes assumption and the h residual principal components are based on non-Gaussian axes assumption. Therefore, △*subregion* can be rewritten by using following equation, i.e.


(a) Original subregions: *subregion*

(b) Reconstructed subregions: *subregion'*

(c) Residual subregions: △*subregion*

Fig. 8. The process for the residual subregion images


(a) $U_{k'}$ of PCA in the subregion spaces

(b) $U_{k''}$ of PCA in the residual spaces

(c) $H_{k''}$ of ICA in the residual spaces

Fig. 9. First row is the first k' principle components in U. Second row is the remaining k'' residual principle components $U_{k''}$. Third row is the independent basis $H_{k''}$ in the residual spaces.

$$\begin{aligned} \Delta subregion &= UU^{T} * subregion - U_{k'}U_{k'}^{T} * subregion \\ &= (U_{k'}U_{k'}^{T} * subregion + U_{h}U_{h}^{T} * subregion) - U_{k'}U_{k'}^{T} * subregion \\ &= U_{h}U_{h}^{T} * subregion \approx U_{k''}U_{k''}^{T} * subregion \end{aligned} \qquad (16)$$

where $U_{k''}$ (see Figure 9.b) is the first k'' (k''=29) principal components in U_h. As a result, the residual subregion weight vector can be calculated by $U_{k''}^{T} * subregion$. In addition, by applying ICA to $U_{k''}$, statistically independent basis images $H_{k''}$ with dimensions k'' can be generated. $H_{k''}$ (see Figure 9.c) is represented by

$$H_{k''}^{T} = T_{k''}U_{k''}^{T} \qquad (17)$$

where $T_{k''}$ is the weight matrix. Bell and Sejnowski's algorithm [2] is used to estimate $T_{k''}$, which is an invertible matrix. Thus, the residual subregions image can be reconstructed by:

$$\begin{aligned} \Delta subregion &= U_{k''}U_{k''}^{T} * subregion = H_{k''}(T_{k''}^{-1})^{T}U_{k''}^{T} * subregion \\ &= H_{k''}(U_{k''}T_{k''}^{-1})^{T} * subregion = H_{k''}B \end{aligned} \qquad (18)$$

Therefore, △*subregion* consists of $(U_{k''}T_{k''}^{-1})^{T} * subregion = B$, which are linear combination coefficients of the independent basis images, $H_{k''}$. Here, the ICA transformation matrix is denoted as $ICA\_TranM_{k''}$ and is computed by:

$$ICA\_TranM_{k''} = (U_{k''}T_{k''}^{-1}) \qquad (19)$$

From equation (3), the Mahalanobis distance $d(I_T)$ becomes:

$$\begin{aligned} d(I_T) &= subregion^{T} * \sum^{-1} * subregion = subregion^{T} * \left[UW^{-1}U^{T} \right] * subregion \\ &\approx subregion^{T} * \left[U_{k'}W_{k'}^{-1}U_{k'}^{T} + U_{k''}W_{k''}^{-1}U_{k''}^{T} \right] * subregion \\ &= y_{k'}^{T}W_{k'}^{-1}y_{k'} + \left(U_{k''}^{T} * subregion \right)^{T} W_{k''}^{-1} \left(U_{k''}^{T} * subregion \right) \end{aligned} \tag{20}$$

where $y_{k'} = U_{k'}^{T} * subregion$ is the weight vector based on eigenvectors $U_{k'}$. The residual subregion weight vector is then transformed to linear combination coefficients of $H_{k''}$ by means of equation (18), i.e., $B = (U_{k''}T_{k''}^{-1})^{T} * subregion$. Therefore, equation (20) for the Mahalanobis distance can be represented as:

$$d(I_T) = \sum_{i=1}^{k'} \frac{y_i^2}{\lambda_i} + \sum_{\substack{i=k'+1 \\ j=1}}^{\substack{i<=(k'+k'') \\ j<=k''}} \frac{B_j^2}{\lambda_i} \tag{21}$$

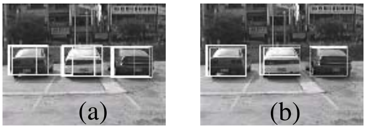


Fig. 10. Vehicle detection without and with position information of the subregions, as show in (a) and (b), respectively

Table 1. The performances of different models evaluated by testing database (a). (PC: P4 3G Hz. 'FA': False Alarm. 'SF': Seconds/Frame.)

	1st M	2nd M	3rd M	4th M	4th M+VQ
R:DR %	87.0%	87.6%	86.6%	92.8%	91.5%
F:DR %	89.1%	89.5%	88.4%	94.0%	93.4%
R:FA	73	53	47	37	46
F:FA	59	43	41	28	44
SF	4.856	1.643	2.455	3.455	0.28

So, the likelihood probability in equation (5) becomes:

$$P(I_T \mid C) = \frac{Exp\left[-\frac{1}{2}\left(y_{k'}^{T}W_{k'}^{-1}y_{k'} + B^{T}W^{-1}B \right) \right]}{(2\pi)^{(k'+k'')/2} \prod_{i=1}^{(k'+k'')} \lambda_i^{1/2}} = \frac{Exp\left[-\frac{1}{2}\sum_{i=1}^{k'} \frac{y_i^2}{\lambda_i} \right]}{(2\pi)^{k'/2} \prod_{i=1}^{k'} \lambda_i^{1/2}} * \frac{Exp\left[-\frac{1}{2}\sum_{\substack{i=k'+1 \\ j=1}}^{\substack{i<=(k'+k'') \\ j<=k''}} \frac{B_j^2}{\lambda_i} \right]}{(2\pi)^{k''/2} \prod_{i=k'+1}^{(k'+k'')} \lambda_i^{1/2}} \tag{22}$$

The posterior probability equation for the 4th model becomes:

$$P(vehicle \mid I_T) = \frac{\prod_{i=1}^{3} P(projection_i^i, ICACoeff_i^i, pos_i \mid vehicle)}{\prod_{i=1}^{3} P(projection_i^i, ICACoeff_i^i, pos_i \mid vehicle) + \prod_{i=1}^{3} P(projection_i^i, ICACoeff_i^i, pos_i \mid non-vehicle)} \tag{23}$$

where $projection_i^i$ is weight vector of the subregion *i* projected into the corresponding eigenspace *i*, and $ICACoeff_i^i$ is the ICA coefficient vector of the subregion *i* projected into the corresponding independent basis *i*.

In the testing process, each subregion of the input window, I_T, is projected into the corresponding eigenspace and the corresponding independent basis space to generate a k'-dimensional weight vector and a k''-dimensional ICA coefficient vector, respectively. Hence, three weight vectors and three ICA coefficient vectors exist for each input window. The posterior probability of the input window, I_T, is calculated from equations (22) and (23). The existence of vehicles can then be determined by assigning different threshold values in equation (9).

3 Experiment Results

A testing database of 457 vehicle images was compiled from the internet and from images captured using handheld video cameras. In total, the database contained 303 rear-viewed vehicle images and 154 frontal-viewed vehicle images. The vehicles in the testing images displayed a wide variety of size and orientation. Moreover, the images featured various background sceneries, lighting conditions and degree of occlusion. In addition, we also tested the following published vehicle databases: MIT CBCL Group 187 rear- and 252 frontal-viewed vehicles images and Caltech Vision Group 526 rear-viewed vehicles images.

Initially, the input image was processed by applying a low-pass filter to remove noises. This image was then down-sampled from original resolution of 240*320 pixels (level 0) to 32*43 pixels (level 15) by a factor of 7/8. In the searching window extraction process, searching window I_T of 32*41 pixels, which is exactly the same size as the vehicle template region, was employed to conduct vehicle detection by shifting this window pixel by pixel at each level.

The non-vehicle information was extracted from the false acceptance subregions by applying the vehicle detection process to the original training vehicle images. The actual vehicle subregion inside the false acceptance vehicle region is not qualified as non-vehicle information. We collected about 10000 images of rear-viewed non-vehicle and 9800 images of frontal-viewed non-vehicle. A similar collection method has been used in [8], [14] and [15].

Figure 10 illustrates the effect of including feature position information in the vehicle detection process. Figure 10.a shows the result of vehicle detection when the feature position information is not considered (1st model). Ignoring this information causes false acceptances between two neighboring vehicles, since subregion 1 encloses the top edge profile of the wall, which resembles the roof profile of a vehicle. In Figure 10.b, the individual vehicles are correctly detected by including feature position information in the detection process (2nd model). The 3rd and 4th models also solve above problem by considering position information.

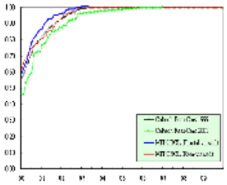


Fig. 11. ROC curves (x-axis is false detection rate and y-axis is detection rate) for vehicle detection using the 4th model.

Table 2. Detection rate comparison using the Caltech rear-viewed vehicle database

	Our 4th model	Fergus, et al. [6]
Detection Rate	92%	84.8%

The four models described in the previous sections were applied to our testing image database. The corresponding experimental results are listed in Table 1. It can be seen that the 4th model yields the best performance, while the 3rd model yields the poorest results. Therefore, the 4th model represents the best approach for vehicle detection. It has the lowest false detection rate and the highest detection rate. The 4th model was then applied to each of the MIT CBCL group and Caltech vision group 1999 and 2001 testing databases. The resulting ROC curves, as shown in Figure 11,

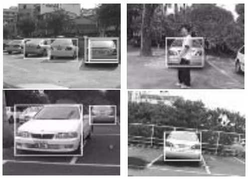
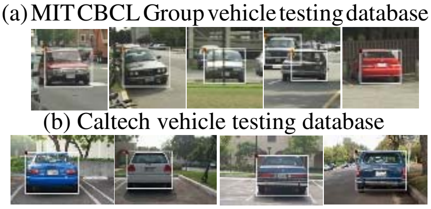


Fig. 12. Tolerances of our vehicle detection



(a) MIT CBCL Group vehicle testing database

(b) Caltech vehicle testing database

Fig. 13. Detection example of using the 4th model

also show consistent and promising performance. Table 2 demonstrates that the proposed system of using the 4th model provides better results for rear-viewed vehicle detection than the method proposed in [6]. In addition, the current vehicle detection system is tolerant to pan and roll rotations, scaling, and partial occlusions, as demonstrated in Figure 12. Some experimental results are shown in Figure 13.

4 Speedup by Vector Quantization

In order to find the maximum posterior probability, it is necessary to compare each weight vector with all the canonical subregions. It is very time consuming (see [15]) because the number of training subregions is huge. To speed up computation, we use vector quantization to classify all the training vehicle and non-vehicle weight vectors into clusters (explained later). Now the comparison occurs between the input weight vector and each of the clustering weight-vector centers.

The training vehicle and non-vehicle weight vectors create two codebooks independently by using the likelihood probability in equation (22) for the measure of the nearest neighbor rule. The initial classification process is only for those weight vectors, whose likelihood probabilities pass the threshold (0.8). Next, the same process and threshold apply on remaining weight vectors started from the center of remaining weight vectors. We repeat the same process until the remaining weight vectors belong to the same cluster or the total cluster numbers do not change. The computational time and result are show in Table 1.

5 Discussions and Conclusion

This study has developed an automatic vehicle detection system based on a statistical approach. Meaningful local features are considered in this detection process. Four possible models for vehicle detection have been proposed in order to overcome the problem of inefficiency associated with traditional methods, and to determine the factors affecting successful vehicle detection. The current experiments have shown that the false alarm rate is directly influenced by the feature position information of the subregions. The 2nd, 3rd and 4th models have lower false alarm rate since they consider the position information of the subregions. The 1st model has the highest false alarm rate because it does not consider the position information of the subregions. It is also found that the detection rate is directly affected by the correlation of the neighborhood pixels, which is a feature of the PCA method. The 1st and 2nd models exhibit similar detection rates because they share the same eigenspace. Meanwhile, the 3rd model yields an inferior detection rate because it uses three individual eigenspaces with wider distribution variances, particularly in subregion 1. This model is sensitive to variations in lighting conditions and vehicle orientation.

The 4th model represents the promising result for vehicle detection. It has the lowest false detection rate and the highest detection rate because the 4th model models parts of each local subregion eigenspace as a Gaussian distribution, while it models residual space as a non-Gaussian distribution. That is, it not only models low frequency information by PCA, but also models high frequency information by ICA applied in the residual space, which can overcome the drawbacks caused by the sensitivity to lighting conditions and vehicle orientation in the 3rd model. Therefore, the 4th model is tolerant of limited pan and roll rotations, and partial occlusion.

References

1. S. Agarwal and D. Roth, "Learning to Detect Objects in Images via a Sparse, Part-based Representation ", IEEE Tran. on PAMI, Vol. 26, No. 11, pp. 1475-1490, 2004.
2. M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski, “Face Recognition by ICA”, IEEE Tran. on Neural Networks, Vol. 13, No. 6, pp. 1450- 1464, November 2002.
3. M. Betke, E. Haritaoglu, and L.S. Davis, “Multiple Vehicle Detection and Tracking in Hard Real Time”, IEEE Intelligent Vehicles Symposium, pp. 351-356, 1996.
4. A. Broggi, “Visual Perception of Obstacles and Vehicles for Platooning”, IEEE Tran. on ITS, Vol. 1, No. 3, pp. 164-176, 2000.
5. F. Dellaert, “CANSS: A Candidate Selection and Search Algorithm to Initialize Car Tracking”, CMU-RI-TR-97-34,1997.
6. R. Fergus, P. Perona, and A. Zisserman, “Object Class Recognition by Unsupervised Scale-Invariant Learning”, IEEE Con. on CVPR, Vol. 2, pp. 264-271, 2003.
7. B. Heisele, P. Ho, J. Wu, and T. Poggio, “Face Recognition: Component-based versus Global Approaches”, Computer Vision and Image Understanding, Vol. 91, No. 1/2, pp. 6-21, 2003.
8. F. Jurie and C. Schmid, "Scale-Invariant Shape Features for Recognition of Object Categories", IEEE Con. on CVPR, pp. 90-96, 2004.
9. M. Kagesawa, S. Ueno, K. Ikeuchi, and H. Kashiwagi, “Recognizing Vehicles in Infrared Images Using IMAP Parallel Vision Board”, IEEE Tran. on ITS, Vol. 2, pp. 10-17, 2001.
10. T. Kato, Y. Ninomiya, and I. Masaki, “Preceding Vehicle Recognition Based on Learning From Sample Images”, IEEE Tran. on ITS, Vol. 3, No. 4, pp. 252-260, 2002.
11. T.K. Kim, H. Kim, W. Hwang, S.C. Kee, and J. Kittler, “Independent Component Analysis in a Facial Local Residual Space”, IEEE Tran. on PR, 37, pp. 1873-1885, 2004.
12. B. Leung, “Component-based Car Detection in Street Scene Images”, Master Thesis, Department of Electrical Engineering and Computer Science, MIT, May 2004.
13. B. Moghaddam and A. Pentland, “Probabilistic Visual Learning for Object Representation”, IEEE Tran. on PAMI, Vol. 19, No. 7, pp. 696–710, 1997..
14. H. Schneiderman and T. Kanade, “Object Detection Using the Statistic of Parts”, Int.l Journal of Computer Vision, Vol. 56, pp. 151-177, February 2003.
15. H. Schneiderman and T. Kanade, “Probabilistic Modelig of Local Appearance and Spatial Relationships for Object Recognition”, IEEE Con. on CVPR, pp. 45-51, 1998.
16. Z. Sun, G. Bebis, and R. Miller, "Object Detection Using Feature Subset Selection", Pattern Recognition Letter, Vol. 37, pp. 2165-2176, 2004.

A Handheld Projector Supported by Computer Vision

Akash Kushal[1], Jeroen van Baar[2], Ramesh Raskar[2], and Paul Beardsley[2]

[1] Department of Computer Science, University of Illinois,
Urbana Champaign, USA
kushal@cs.uiuc.edu

[2] Mitsubishi Electric Research Laboratories, USA
{jeroen, raskar, pab}@merl.com

Abstract. This paper describes the use of computer vision to support the operation of a handheld projector, and describes four applications. Projectors in the past have been used as fixed devices, but the latest generation of 'pocket projectors' is small and portable. We demonstrate the feasibility of using a projector held in the hand, and the types of applications that can be done with a handheld projector.

We attach a camera to the projector to support its operation in two ways. Firstly, vision is used to recover the motion of the projector relative to the display surface. A handheld projector with motion recovery allows a range of interesting functionality, and we show how web-browsing can be done with a handheld projector, complete with mouse interaction and text-entry. Secondly, we use the camera to process information about the projection surface - for example we demonstrate an application that allows a user to attach digital information to a physical texture, and later to recover and view the digital data via recognition of the texture.

1 Introduction

One of the most desirable characteristics for a handheld information device is small size, but the drive to reduce size is in direct conflict with the need for the device's display to be large enough to convey a useful amount of information. An illustration of the problem is that while cellphones are shrinking, they cannot effectively support an application like web-browsing because cellphone screen size is insufficient for web pages that have been created for screens of 15 inches and up. At the same time, projectors are becoming smaller, and several different manufacturers have introduced 'pocket projectors'. Figure 1-left shows a model by Mitsubishi, which weighs 14 oz and fits in the hand. The small size makes it easy to transport, and with a battery life of a few hours, the projector is now becoming a personal portable device just like the laptop.

Now consider a cellphone augmented with a projector. Anticipating that projectors will continue to decrease in size, the device could potentially be very small, but it can create a projection that is similar in size to physical desktop and laptop displays - thus it is possible to have a small device *and* to handle an information-heavy application like web-browsing. This idea motivates the work

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 183–192, 2006.

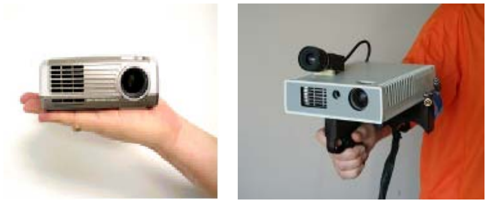


Fig. 1. At left, a commercial ‘pocket projector’; at right, our current prototype handheld projector with camera and grip

here. Of course, a handheld projector must not only display information, but also allow ways to interact (mouse, text-entry). We address both aspects in this paper.

A key technology for a handheld projector is a method for recovering the motion of the projector relative to the display surface. We use computer vision. The primary reason is that vision is versatile, allowing us to develop a range of motion-recovery algorithms for specific applications. In addition, a camera is a cheap component to add to a handheld device. Note that our goal is to create a *self-contained* handheld projector. There are other ways to provide the functionality in this paper if there is fixed infrastructure in the environment (e.g. see [1] for laser pointer interaction in fixed installations). Fixed infrastructure is fine for some applications but our aim is to develop a self-contained device.

Figure 1-right shows the prototype handheld projector. Its components are (a) a Plus V-1080 projector, 1024x768 pixels, 60Hz, (b) a Basler A602F camera, 640x480 pixels, 100Hz, (c) four rigidly attached laser pens, two on either side of the case, (d) a hand-grip on the base with a click button under the index finger for input, (e) umbilical to a computer. The device weighs about 2.5lb so it is heavy for extended use, but it has been suitable for our experiments so far.

Contributions: Previous work on projector-camera interfaces includes [2], and work on steerable projectors includes [3][4]. This paper builds on existing work in [5],[6] and extends it by including (a) handheld projection on display surfaces having an unknown texture, (b) a method for text input and accompanying applications, (c) a ‘light stylus’ application.

2 Calibration

We do a full euclidean calibration of the projector, camera, and four laser pointers. All references to a ‘display surface’ below and in the rest of the paper imply a *planar* display surface. The **projector-camera calibration** is as follows

- Camera intrinsics K_c: we take several distinct views of a planar calibration pattern and use a standard calibration method.
- Projector intrinsics K_p: we take several distinct views of a planar calibration pattern while simultaneously using the projector to project a distinct pattern onto it. The camera simultaneously observes the physical calibration

pattern and the projected pattern. Thus we can infer euclidean coordinates for the projected pattern. The problem now reduces to a standard calibration procedure, because we have projector pixel coordinates plus corresponding euclidean world coordinates, for several views of a pattern on a plane. An alternative way to compute projector intrinsics, which avoids the use of a physical calibration pattern, is described in [7].

- Projector-camera extrinsics R, T: we take several distinct views of a blank display surface while using the projector to project a pattern onto it. We collect point correspondences between the camera and projector image planes and compute the fundamental matrix F_{cp} between the camera and projector. We compute the essential matrix $E_{cp} = K_c^t F_{cp} K_p$, and decompose to R, T using a linear computation, followed by a nonlinear computation that minimizes an image plane error [8].

The **laser calibration** is as follows. The laser pens create rays in space. We wish to compute (a) euclidean 3D line equations Q_i of the four laser rays in the projector-camera coordinate frame, (b) for each ray, the projections l_i^c and l_i^p of the ray onto the camera image plane and projector image plane respectively, (c) for each ray, the line homography L_i^{cp} that describes the mapping of points between l_i^c and l_i^p. (The use of this information will be described in subsequent sections). The approach is

- Take an image of a blank display surface while the projector is projecting a pattern, and while the four lasers are projecting four laser spots.
- Detect the pattern, and compute the homography H_{cp} from the camera to the projector image plane.
- Detect the laser spots s_i^c on the camera image plane. For each laser spot, compute $s_i^p = H_{cp} s_i^c$, the inferred position of the laser spot on the projector image plane. Store the correspondence (s_i^c, s_i^p).
- Repeat for three (or more) distinct views.
- Compute the best-fit straight line l_i^c from the stored points s_i^c for the three views. Similarly, compute l_i^p from the stored points s_i^p for the three views. Compute the line homography L_i between l_i^c and l_i^p using the stored correspondences (s_i^c, s_i^p) for the three views.
- Reconstruct the 3D line equations Q_i of each laser ray using the projector-camera calibration and the lines l_i^c and l_i^p.

3 Basic Functions

This section describes the basic functions of the handheld projector - stabilizing a projection so that is fixed on the display surface, our method for doing mouse input with the projector, and use of the lasers to improve the processing.

Stabilizing the Projection: the first goal is to create a stabilized projection (the projection image is fixed on the surface, even when the projector is moving), which is keystone-corrected (the projection image is a rectangle of the desired aspect ratio, even if the projector is skew to the surface). Consider the simplest case. We have four markers $M_1...M_4$ on the display surface that form a rectangle with the same aspect ratio as the desired projection image. Call the four vertices of the projection image on the display surface $V_1...V_4$. The method to stabilize the projection so that it appears fixed within the marked area is

- Detect $v_1^c...v_4^c$, the projections of $V_1...V_4$ on the camera image plane. Knowing the corresponding projector pixel coordinates for these vertices, compute H_{cp}, the homography between the camera and projector induced by the display surface.
- Detect $m_1^c...m_4^c$, the projections of $M_1...M_4$ on the camera image plane.
- Compute $m_i^p = H_{cp}m_i^c$, $i = 1..4$, the inferred projections of $M_1...M_4$ on the projector image plane.

Knowing the projection of $M_1...M_4$ (the desired projection area) on the projector image plane, it is straightforward to warp the projector image so that it projects to the desired physical position. The whole process is repeated at each new time-step. Figure 2 shows two example time-steps.

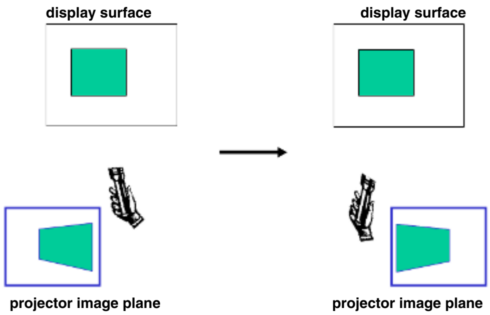


Fig. 2. Stabilization - the projector image plane is continually updated so that the projection on the display surface remains fixed

Doing Mouse Input with a Projector: we make a single modification to the processing above to do mouse input with the projector - the center pixels of the projector image plane are set at each time-step to show a cursor graphic. The effect of this is that the user sees a stabilized projection on the display surface *plus* a cursor that tracks across the projection in direct correspondence with the projector motion. Once the cursor is at the desired location, items are selected in the usual way by clicking a button on the handheld projector. We can now replicate all the familiar mouse interactions (selecting drop-down menus, clicking buttons, scrolling, dragging) within the projector domain. Figure 3 shows two example time-steps.

We adopt this approach in preference to a touch-pad mouse because (a) a touch-pad would add bulk to the device, (b) it would require two-handed use, with context switching between the device itself and the display surface, and (c)

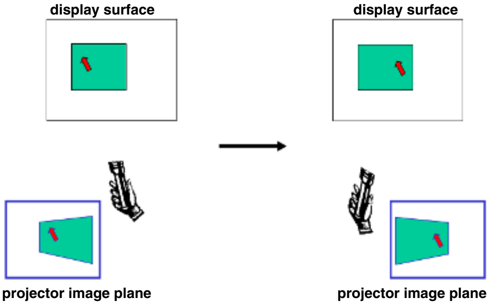


Fig. 3. Mouse interaction - adding the cursor graphic at a fixed position in the center of the projector image plane results in a cursor that tracks across the stabilized projection on the display surface in direct correspondence with the user's pointing motion

it is hard to do fine control of a cursor on a large projected area using a small touch-pad.

Using the Lasers to Compute H_{cp}: Part of the processing in the stabilization was to compute $v_1^c ... v_4^c$ with the ultimate goal of computing H_{cp}. But automatically detecting v_i^c (the camera view of the current projection on the display surface) might be unreliable for some projected images. This section describes how we use the (easily detected) laser spots to compute H_{cp}, avoiding the need for v_i^c.

- Detect $x_i^c, i = 1..4$, the projections of the laser spots on the camera image plane. (The projected laser spots x_i^c are constrained to lie on the lines l_i^c that were computed in Section 2, so identifying them is especially easy).
- Compute $x_i^p = L_i x_i^c$, $i = 1..4$, the inferred projections of the laser spots on the projector image plane, where L_i are the line homographies computed in Section 2.
- Compute H_{cp} using the four correspondences (x_i^c, x_i^p).

4 Handling Display Surfaces with Unknown Texture

The previous section described stabilized projection and interaction on a display surface that had known euclidean properties. This section addresses stabilization and interaction on a display surface that has an unknown texture. First assume four distinct points $N_1 ... N_4$ on the display surface. Four points are sufficient to define a projective coordinate frame on the surface. We can define a desired location for the projection in this coordinate frame, and as long as the points are being tracked, we can project to the same fixed position. Furthermore we can readily upgrade to a euclidean coordinate frame (to support keystone-correction) because the handheld projector is calibrated. The approach to initialize the processing is

- Detect $x_i^c, i = 1..4$, the projections of the laser spots on the camera image plane.

- Compute the 3D coordinates of the laser spots, and then compute the 3D coordinates Z for the plane of the display surface.
- Detect n_1^c, n_2^c, the projection on the camera image plane of two arbitrary points N_1, N_2 on the display surface. Backproject n_1^c and intersect with Z to define the origin of the coordinate frame; backproject n_2^c and intersect with Z to define the unit point on the x axis. Hence obtain a euclidean coordinate frame.
- Select the desired location, vertices D_i, for the projection image, in the coordinate frame on the display surface.
- Project D_i to pixel positions d_i^c on the camera image plane.

The approach to propagate the coordinate frame at each time-step, and to display the projection image, is

- Track features $n_i^c, i >= 4$ between the previous frame $j-1$ and the current frame j, hence obtain feature correspondences between frame 0 and frame j, and compute the homography T_j between the frames 0 and j induced by the display surface.
- Propagate the euclidean coordinate frame to the current camera image by using T_j to transform d_i^c.
- Use H_{cp} to transform d_i^c to the projector image plane. Knowing the coordinates of the desired projection location on the projector image plane, it is straightforward to warp the projector image so that it projects to the desired physical location.

Texture Tracking: Matching between frames employs a global scheme that searches for a consistent transformation over the matched features. The process is initialized with the set of features detected using a Harris corner detector in the base frame (frame 0). For each subsequent frame i we compute the homography T_i between the base frame and the ith frame. To compute T_{i+1} we first transfer all the features in the base frame to frame i using T_i. Then, we search in a small window around each transferred feature for its matching feature in frame $i+1$. The candidate matches are filtered by an acceptance threshold on the normalized correlation between the matched features in the ith and $i+1$th frame, and we use RANSAC to identify a set of matches consistent with a homography. The matches are used to compute T_{i+1}. New features that get detected in later frames are transformed from their current frame to base frame and added to the set of base features, to allow the projector to move away from the initial base frame position without losing the tracking.

5 Applications

We propose the following taxonomy of applications for a handheld projector (a) applications that use a display surface where the only texture consists of markers to guide the projection - see Section 5.1, (b) applications that use a display surface with an unknown texture - see Section 5.2 and 5.3, (c) applications that project augmented reality onto a known object - see Section 5.4. There are no quantitative results below. The calibration procedures are a variation on known results for pure camera systems, and there are few observations to be made - the epipolar geometry errors are about 0.5 pixels for the camera-projector and are similar to what one would obtain with two cameras.

5.1 Web-Browsing

Figure 4a shows a projection of a live application - the Google web page. The standard web page is augmented with a 'Text' button at lower-right which the user presses to initiate text-entry. Figure 4b: after initiating text-entry, the user holds down the handheld projector's click button and forms a letter 'v'. We use *libstroke* for stroke recognition [9]. Figure 4c: after completing the letter, the user releases the click button and is presented with the recognised letter. The same letter is sent to the text-field that is currently active on the web page. After completing text-entry, the user clicks the 'Text' button again to return the mouse to normal cursor mode.

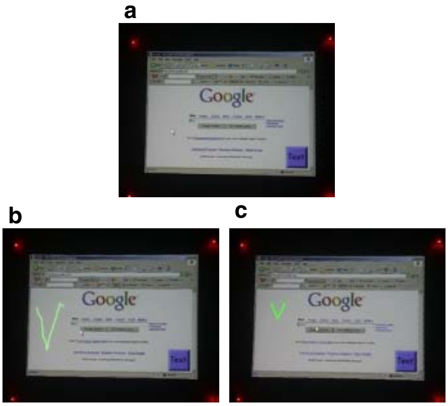


Fig. 4. Web-browsing application including text-entry

5.2 Light Stylus

Laser pointers are commonly used to indicate a point of interest on a big-screen slide presentation. We extend this functionality to allow a user to create arbitrary doodles on the slides, such as underlines, arrows, or circlings around areas of interest. We call this a 'light stylus'. There are ways to achieve this functionality that make use of fixed infrastructure in the environment, but our approach is a completely self-contained, portable device, making it more flexible in a variety of settings. Figure 5a shows an example of a big-screen slide presentation from a fixed projector. Figure 5b: the user directs the cursor to the start point of an underline on the text, presses the click button, then directs the cursor to the end point of the underline and double clicks. The underline is subsequently

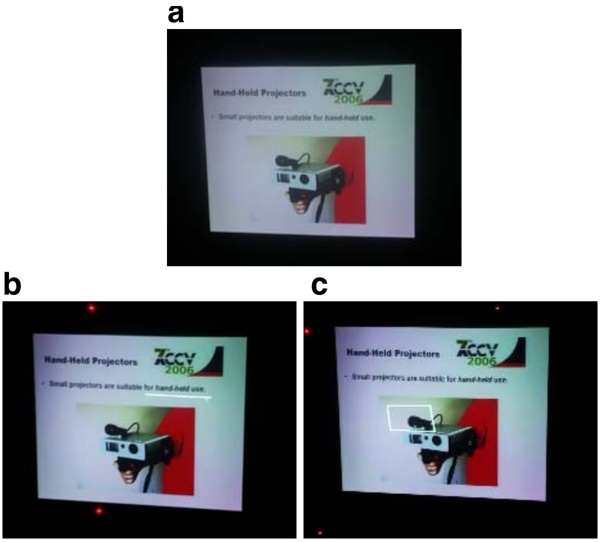


Fig. 5. Light stylus application showing underlining and drawing a box on a slide presentation

shown at the specified position until the next interaction. Figure 5c: the user employs the same interaction with a series of four clicks to draw a box around an object on the slide. **Beware a possible confusion -** the slide presentation is created by a fixed projector, such as one finds in a conference room, *not* by the handheld projector; the handheld projector is used only to create the augmentations in Figures 5b and 5c. This application runs on any texture, not just a slide presentation, and it has also been used to do underlining and circling on posters.

5.3 Electronic Sticky Notes

This application demonstrates how to attach digital information to some physical texture (a CD case), and later retrieve the information automatically the next time that the CD case is seen. Figure 6a: the scene consists of some random objects and a CD case. The user directs the cursor to a start point near the CD case, clicks to start a selection, then defines the (projected) blue polygon by clicking at each vertex, and double-clicking at the final vertex. For clarity, we refer to the part of the camera image within the blue polygon as a *texture-key.* The texture-key is stored along with a user-specified text-entry 'Return Date: 25th Jul'. Figure 6b: the user directs the handheld projector at a new scene containing the CD case, and requests a retrieve operation. The image is matched against all stored texture-keys. If a match is obtained, the corresponding text-entry is projected next to the recognised object - in this case, the text 'Return Date: 25th Jul' is shown next to the CD case. As an aside, note that projection onto darkish wood is clearly visible. We match the image against stored texture-keys by feature matching, where the features are pixel patches around corners.

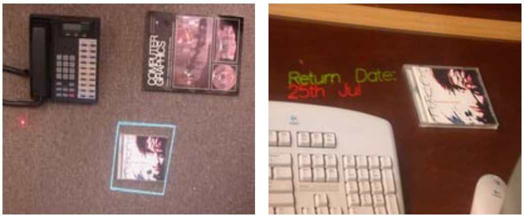


Fig. 6. Electronic sticky note - (a) object selection for attaching a note, (b) retrieval of note

This is fine for small databases of texture keys and small change in view direction, but of course we would need more sophisticated methods for a truly practical system.

5.4 Projected Augmented Reality

A key application of the work is projected augmented reality as a way to interface to physical devices that can wirelessly communicate their internal state. The handheld projector retrieves the state, projects it next to the device, allows the user to interact to specify a desired operation, and then transmits the operation back to the device. In this way we can provide complicated control panels for physical devices, without the device needing any sort of physical display or physical input device. See Figure 7 for an example of projected augmented reality.

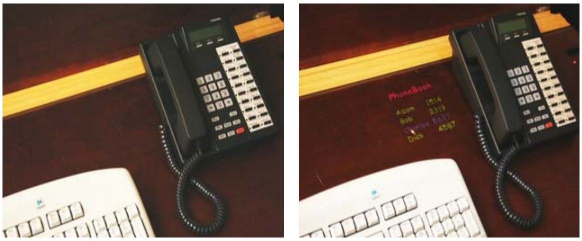


Fig. 7. Projected augmented reality - projecting a phonebook next to a recognised phone

6 Conclusion

This work is speculative. Based on our experience with a simple game application, which was tried by hundreds of casual users over several days, there is not a problem with its usability. People were able to guide the cursor and to click and drag objects, and while there were comments about the weight of the prototype, many people seemed to feel a real sense of interacting with a projection. But this type of unusual device still raises questions about practicality. Assuming handheld projectors do appear, aren't there easier ways to interact with a projection? It's true that one could attach touchpad or thumbwheels for a familiar type of mouse interaction, but consider how simple and direct the approach in this paper is - one-handed pointing of the projector to guide the mouse. Secondly isn't the current device unwieldy? Our latest generation device has a projector half the weight of the current one, and the trend to more compact projectors is continuing. Isn't the technique wasteful of projector pixels because it uses only part of the projector image plane for the stabilized image? Just as projectors are continuing to become lighter, so the number of pixels continues to increase. And even using a limited part of the projector image plane, we have sufficient pixels for the applications described. In summary, this a workable idea that provides novel functionality for the fast-approaching situation when projectors are incorporated in handheld devices.

References

1. Dan R. Olsen, Jr. and Travis Nielsen, "Laser pointer interaction," in *Proceedings of the SIGCHI conference on Human factors in computing systems.* 2001, pp. 17–22, ACM Press.
2. R. Sukthankar, R.G. Stockton, and M.D. Mullin, "Automatic keystone correction for camera-assisted presentation interfaces," in *Proc. Intl. Conf. Multimedia Interfaces '00*, 2000.
3. Claudio Pinhanez, "Using a steerable projector and a camera to transform surfaces into interactive displays," in *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, New York, NY, USA, 2001, pp. 369–370, ACM Press.
4. M. Ashdown and Y. Sato, "Steerable projector calibration," in *Proc. IEEE Workshop on Projector-Camera Systems*, 2005.
5. R. Raskar, J. VanBaar, P.A. Beardsley, T. Willwacher, S. Rao, and C. Forlines, "iLamps: Geometrically aware and self-configuring projectors," in *SIGGRAPH 2003 Conference Proceedings*, 2003.
6. Paul A. Beardsley, Jeroen Van Baar, Ramesh Raskar, and Clifton Forlines, "Interaction using a handheld projector," *IEEE Computer Graphics and Applications*, vol. 25, no. 1, pp. 39–43, 2005.
7. T. Okatani and K. Deguchi, "Projector-screen-camera system: Theory and algorithm for screen-to-camera homography estimation," in *Proc. International Conference on Computer Vision*, 2003.
8. R.I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
9. http://www.etla.net/libstroke/, "Libstroke - a stroke translation library," .

FormPad: A Camera-Assisted Digital Notepad

Tanveer Syeda-Mahmood and Thomas Zimmerman

IBM Almaden Research Center, 650 Harry Road, San Jose, CA, USA 95120
{stf, tzim}@almaden.ibm.com

Abstract. A camera-assisted digital writing tablet was invented recently. It preserves the familiar experience of filling out a paper form while allowing automatic conversion of relevant handwritten field entries into electronic form, without explicit form scanning. In this paper, we focus on two key computer vision problems associated with the invention of this device, namely, form indexing and field projection. These are needed for accurate association of tablet writing with corresponding entries in the electronic form. Form indexing is modeled as the problem of shape-based content retrieval using the perspectively-distorted form appearances seen from the tablet camera. Fast form indexing is achieved using geometric hashing based on projective invariants. The invariants derived from curve and line features reduce the basis search space considerably while still providing for robust localization. We derive field projection as a sequence of projective transformations between the tablet, the camera and the original electronic form coordinates. Results of extensive testing on a medical form database are reported.

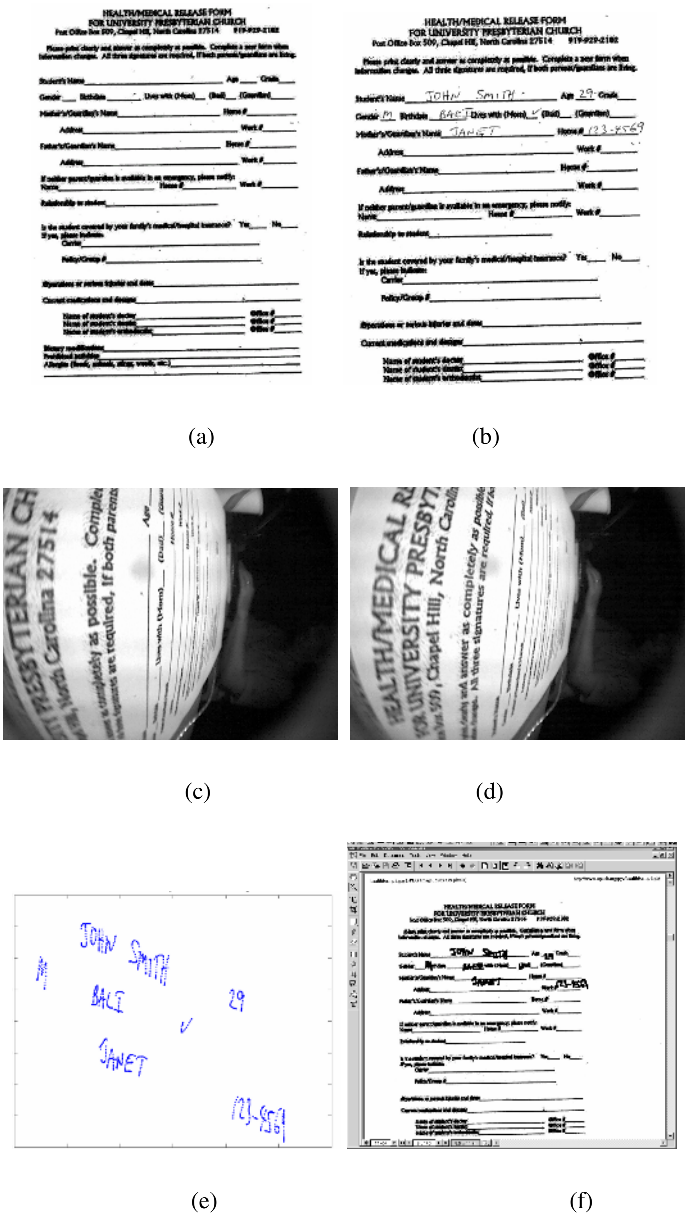
1 Introduction

Paper-based forms are ubiquitous in hospital environments. With high volume of forms being scanned, and the difficulty of handwriting recognition from filled form entries, most electronic record systems simply store the form images with the field label information entered manually. A camera assisted writing tablet called the FormPad was invented recently to enable direct electronic conversion of form entries. Unlike other digital notepad-like devices such as the CrossPad, FormPad has the ability to recognize the form and accurately project the filled entries against their correct field label. Such digital notepads are a low-cost alternative to tablet PCs for routine form filling. They also preserve the familiar experience of filling in a paper form without disturbing the existing workflow of end users.

The FormPad device is a conventional clipboard with a pen digitizing tablet [8] underneath and a VGA digital camera [9] with fish-eye lens (64 x 86 degrees) attached to the metal clip of the clipboard. A wireless inking pen allows the user to enter notes directly on the form, while the digitizer captures pen coordinates and pen tip pressure. Thus form filling actions are recorded as online handwriting signals by the tablet. In order to use the data from the tablet, however, the identity of the form being filled must be known. Further, the handwritten data must be correctly registered against the relevant entries in the electronic form.

Accurate form identification and field projection using cameras is a challenging problem. Ease of use considerations require that the camera be placed in un-obstructing locations on the notepad leading to significant perspective distortion in the captured

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 193 – 203, 2006.


HEALTH/MEDICAL RELEASE FORM
FOR UNIVERSITY PRESBYTERIAN CHURCH
Post Office Box 509, Chapel Hill, North Carolina 27514 919-929-2102
Student's Name JOHN SMITH Age 29 Grade
Gender M Birthdate BACI Lives with (Mom) ✓ (Dad) (Guardian)
Mother's/Guardian's Name JANET Home # 123-4569
JOHN SMITH
M
BACI
29
✓
JANET
123-4569

(a)

(b)

(c)

(d)

(e)

(f)

Fig. 1. Illustration of field projection of tablet writing. (a) Original form. (b) A filled form. (c) The reference model for the form of (a) as seen through camera. (d) The camera appearance of the form of (b) before filling. Note the skew in this appearance. (e) Tablet writing corresponding to the filled entries (f) Tablet writing projected into the electronic form of (a) using our method of field projection.

images. Also, since the camera is very close to the imaged object (i.e. the form), weak perspective projection models do not hold, requiring the use of full projective transforms. Existing method for recognition under perspective projection, are either compute or space intensive requiring at least 5 point correspondence raising the complexity to O(N^5). Even after the correct form image has been identified, pose registration errors, if present, can lead to the tablet data being recorded against the wrong field label in the electronic form. Thus careful analysis of the geometric relationships between the tablet, the camera and the electronic form coordinates must be performed.

2 Related Work

The technology we exploit in FormPad is based on prior work on object indexing and form recognition, for which a large body of literature already exists. In particular, recognition of scanned forms has been addressed by a number of researchers [6, 7, 10, 11, 12-16], and the technology has matured into many products including AccuForm, CharacTell, iRead, ReadSoft, etc. Several low-level form processing and feature extraction methods [10, 11] exist, including those that analyze layout [7, 17], fields [14], and hand-filled entries [11, 12].

Registration methods based on projective geometry have been used for scanned form alignment and recognition [15]. While almost all form recognition work assumes scanned forms, the only significant work on camera-grabbed forms we found was the document imaging camera system ScanWorks by Xerox [16]. The focus in this system has been on image processing of the document to filter, de-skew and produce better document appearance rather than form identification and automatic field extraction. The predominant techniques for identifying the form type use bar codes or OMR technology. The recognition of printed text on forms is done fairly well using commercial OCR engines and most OCR software also offer their engine bundled in form recognition software. The recognition of handwritten text, however, is still a difficult problem for scanned forms.

The work on form indexing we report is based on the technique of geometric hashing previously introduced for the model indexing problem in computer vision [2]. Several variants of this technique have appeared in literature including line hashing [3] where the basis space was formed from lines, location hashing [4] and region hashing, hashing based on projective invariants [1], etc. While geometric hashing using affine-invariant features has found some practical applications, much of the work on geometric hashing using projective invariants has remained mostly academic in nature. Building practical embedded systems using such techniques has been a challenge due to the large number of combinations of basis features that need to be retained per model, and their sensitivity to noise and occlusions. Thus building practical form recognition systems using geometric hashing requires intelligent choice of basis features that reduce the time and space complexity while still giving effective recognition.

3 Form Recognition

We now turn to the problem of form identification, which can be stated as follows. Given a sample form C’ seen by the FormPad camera, determine the original form O corresponding to C’ using the appearance form images in the database C. In practice, since the number of forms in the database is large, and live form processing is desirable (as manual on-the-spot correction of form entries by the FormPad user may be required), it should be possible to identify the original form without exhaustively searching the form database.

To recognize the original form O corresponding to the given sample form on the tablet, it is sufficient to determine if the associated reference form C in the model database and C' are two views of O. Since forms are planar objects, and since the distance between the camera and the form is smaller than the form dimensions, the relation between the two views C and C' is a projective transform P. That is, given a point (x,y) in C' its corresponding point (x',y') in C is related by

$$x' = \frac{p_{11}x + p_{12}y + p_{13}}{p_{31}x + p_{32}y + 1}, \quad y' = \frac{p_{21}x + p_{22}y + p_{23}}{p_{31}x + p_{32}y + 1} \tag{1}$$

where the coefficients are elements of the projective transform P given by

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & 1 \end{bmatrix} \tag{2}$$

It is well known that the above 8 parameter projective transform for planar objects can be recovered from a set of 4 corresponding points through a linear system of equations.

Once the projective transform is recovered, it can be verified by projecting the rest of the sample form features into the model form, and noting the fraction of sample form features that fall near the model form features.

3.1 Form Hashing

Because of their text and graphical content, form images tend to have a large number of features, for example, 3000 corners and 2000 lines. If each model form in the database was exhaustively searched, this would take O(m4 n4 *N) time where m and n are the average number of features per model form and sample form respectively, and N is the number of forms in the database.

The relevant forms can be identified without detailed search using the principle of geometric hashing for form indexing. The basic principle is well-known, and involves recognizing an object by verifying that enough number of object features have the same pose-invariant coordinates with respect to some chosen basis frame [2]. Detailed search is avoided by pre-computing pose-invariant feature information, and indexing the recorded data structure using pose-invariant features derived from the current form on the tablet. To provide robustness to occlusions and noise, many more basis frames may have to be used, leading to a large number of redundant features. Much of the space complexity of indexing by geometric hashing is accounted by these additional basis frames and the pose-invariant features so derived.

Our form indexing is based on the area cross-ratio, a projective invariant given by

$$\rho(A,B,C,D,E) = \frac{P(A,B,C)P(A,D,E)}{P(A,B,D)P(A,C,E)}, \tag{3}$$

where A,B,C,D are the basis frame and E is a new point on the planar object. P(A,B,C) is the area of the triangle with vertices A, B and C as shown in Figure 2a.

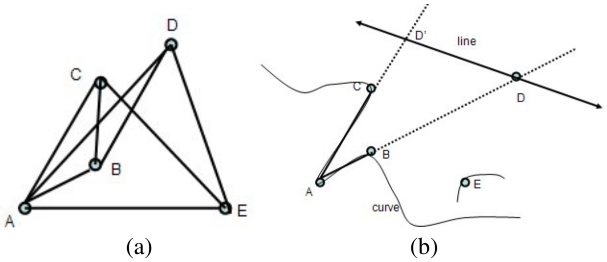


Fig. 2. Illustration of 5 point cross-ratios. (a) Cross-ratios from arbitrary 5 points. (b) cross-ratios from carefully chosen basis frames derived from curves and lines.

If we retain all possible basis frames to provide robustness to noise and occlusions, the space complexity of hashing is very large. In fact, for N=500 corner features on the object, the basis frames and projective invariants computed would be O(N^5) or 1000 Terrabytes, a very large hash table indeed! Furthermore, by choosing features from all over the form image to form the basis frame, the chance of false positives increases with many spurious matches. Both these issues can be avoided if we generate the basis frames carefully to reduce the number of basis frames, and choose at the same time, features that can capture shape-specific information better. For this reason, we choose features from curves to form basis frames. Curves are well-known grouping units that capture shape information present in the model object. Further, due to the order present in curves, the number of basis features can be reduced. Using this rationale, we generate the basis frame as follows.

3.2 Basis Frame Generation

We take corner features from curves to form candidates for point A in the basis frame. Feature points B and C are then taken to be the adjacent corners in the ordered sequence of features along the curve. A fourth point is derived by the intersection of a line anywhere in the image as shown in Figure 2b. Note that at least one intersection point D or D' exists since the line cannot be parallel to both lines emanating from a corner. Since there are two possible directions along which a curve can be traversed, we consider both intersection points D and D', if available to form two sets of basis frames. (A,B,C,D) and (A,B,C,D'). Any new feature point E can now be expressed in terms of the basis frame (A,B,C,D) through its projective invariant as defined in Equation 3. The labeling of features points uses the convention of A for a corner in the curve, B and C for the adjacent corners on either side, and D and D' stand for the intersection of a line with AB and AC respectively. Thus the issue of permutation of feature points leaving ambiguity in matching, is reduced using this naming convention.

Use of consecutive features along the curve makes the choice of 3 features of the basis frame linear in the number of features along the curve. However, it can be too narrow a basis frame when the features are close together leading to instabilities in pose computations. By choosing a fourth basis point from an arbitrary line in the image, the basis frame is widened to allow robust computation of projective invariants.

Using this method of basis frame generations, the number of basis features is $O(N^2)$ with the total number of projective invariants using all features in the image to be $O(N^3)$. Using 500 features as the typical example as before, the size of the hash table per form will now be $O(10^9)$ or 1Gbyte, a reduction by a factor of 10^6!

3.3 Form Indexing

Form hashing involves two stages, namely model creation, and form indexing. In the model creation stage, curves are extracted from reference form images using a technique described in an earlier work [4]. Points (corners) and lines are extracted from the curves through line segment approximation of curves. Basis frames are generated as described above, and projective invariant of Equation (3) is computed for all corner features in the form images. In generating the basis frame, we traverse the curves in both directions to account for reversal of ordering during query processing. The resulting information is recorded in a hash table as

$$H_1(\rho) = \{< Basis_i, F_j > ......\} \tag{4}$$

where $Basis_i$ = <A,B,C,D> are the basis frame coordinates and Fj is form index.

For a given sample form in the tablet, features are extracted through a similar process and candidate basis frames are derived. The projective invariants computed for all feature point on the sample form are used to index the hash table and a histogram of basis indexes is taken. The form index corresponding to the peaks identifies the relevant form. Since hashing indicates likely matches, detailed verification step is still needed to confirm the presence of a reference form using the 4 point correspondences generated from the matching basis frames. The fraction of sample form features that project close to a model feature constitute the verification measure. Such features can also be taken as additional corresponding points for robust computation of the actual projective transform of Equation 1 for form registration later. Although the same cross-ratio can be derived from many combinations of basis frames, using the constrained basis generation process ensures that hashing points to related shapes in the form database.

4 Field Projection

We now turn to the field projection problem which is illustrated in Figure 3. Figure 3e shows the handwriting recorded on the tablet in terms of tablet coordinates. The actual filled form is shown in Figure 3b. The electronic form is shown in Figure 3a. In order to capture the handwritten entries against the appropriate field label in the electronic form of Figure 3a, we need to project the fields as recorded in tablet writing of Figure 3e into the electronic form of Figure 3a. Since the form on the tablet can be placed with skew, it is not possible to do the field projection of the tablet coordinates without using camera-generated information.

Our method of field projection exploits the geometric relationships between the tablet coordinates, the form coordinates in the camera and the original electronic form coordinates. Let O be the original electronic form. Let C be the image of the printed version of this form as seen through the camera on FormPad when placed within fixed alignment reference markers for model generation. Let T be the tablet frame. Let C' be the image of a sample (possibly inserted with a skew) form that is currently being filled by a user. The problem of field identification is to convert the tablet coordinates corresponding to the image C' and rendered into the original form coordinates O. As shown in Figure 3, the projection of a handwriting coordinates (xt',yt') into their corresponding field location (x0,y0) on the original form involves a sequence of transforms T'->C'->C->T->O.

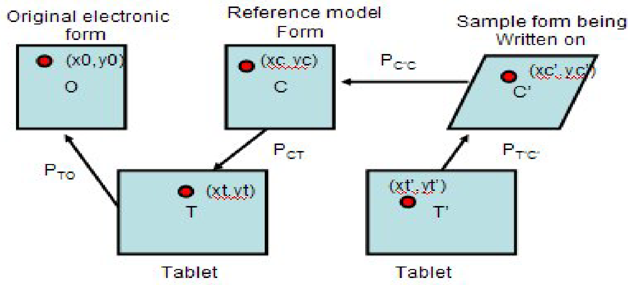


Fig. 3. Illustration of the sequence of transformations needed for field projection

The relationship between tablet coordinates and original form coordinates (T->O) can be modeled by an affine transform P_{TO}. For all electronic forms of the same size, say, 81/2x 11in, and using a systematic generation of the original form image (by pdf to tif conversion software, for example), such a transform need only be computed once. To use this transform directly for any paper form placed on the tablet though, we print the electronic form, and place it on the clipboard within fixed reference markers. This ensures that all forms will be subject to the same reference model creation process, and allows the use of a single alignment transform from tablet to original form.

In camera coordinates, the form skew (C' -> C) can simply be estimated by the projective transform $P_{C'C}$ computed during the form indexing process. Because of the close positioning of the camera on the tablet, the relationship between the camera and the tablet (T->C) is also modeled by a projective transform P_{TC} from tablet-to-camera and another projective transform P_{CT} from camera-to-tablet (C->T). Since the camera is fixed, these are computed only once per tablet during a factory calibration stage.

The overall transformation can thus be modeled as a sequence of projections

$$P_{TC} - > P_{C'C} - > P_{CT} - > P_{TO} \tag{9}$$

of which, only the transform $P_{C'C}$ needs to be computed dynamically per form filling. Note that it is not required for the handwriting on the tablet to be visible to the camera since the camera-to-tablet transform is pre-defined and can be applied as long as the reference model creation process was consistent using the alignment markers.

5 Results

Form indexing and field projection was tested on a large medical form database. These forms were collected from actual forms used in hospitals, as well as those available on the internet. The current collection has 180 electronic forms and 200 scanned forms and is growing rapidly. Models of the forms were created by placing the forms on the FormPad, as well as screen grabbing the electronic documents to make the original form images. Some original forms were obtained by scanning the printed forms available. To test form indexing using form hashing, we recorded 5 -13 different appearances for each of the forms assembled above to increase our model database size to 1800. The form images were processed for edge detection using Canny edge detector. Curves were extracted using a procedure previously described in [4], and corners and lines were assembled. The basis generation process described earlier, was used to record the hash tables entries for form indexing.

We tested the performance of form indexing by querying 40 sample forms on the 1800 form model database. The precision recall results are shown in Table 1. As can be seen, form indexing retains good identification accuracy while still containing the number of false positives. It can also be seen from this table that retaining only well-separated basis frames has not degraded the recognition performance. Our experiments have revealed an average precision of 75.21% and an average recall of 92.5%.

Table 2 shows the time performance of form hashing in comparison to actual search for matching triples during object recognition. As can be seen, pre-computing the features improves the time performance by several orders of magnitude. The average number of features on the model was 424.62 for our model database. The column on all possible basis triples uses O(N^5M^5) for its calculation for model and image features for a straw-man comparison. The search is listed in terms of number of basis triples explored, since the number of pose-invariant features computed is the same in both approaches.

By using well-separated basis frames derived from curves instead of all possible basis frames, the storage performance was improved remarkably. With the average model features of 424.62 the size of the hash table was roughly 424.62*424.62* 424.62 or 1G. As a result, the hash table was stored in main memory itself for all the 1800 forms in the model database.

5.1 Field Projection Results

To test field projection, the sample forms were derived from the electronic forms whose models were already available. Twenty subjects were recruited for writing on the sample forms using the FormPad. The subjects were asked to follow their normal writing process as if on a clipboard. Thus considerable skew could be present in the

Table 1. Illustration of Precision-Recall for Form Indexing

Query Form	Actual Occurrences	Matches Retrieved	False Matches	Correct Matches
1	10	13	5	8
2	4	7	3	4
3	8	10	3	7
4	13	13	3	10
5	9	14	6	8
6	11	15	4	11

Table 2. Time reduction due to indexing results

Query	Query Features	Retained Features	Retained Triples	Search All Possible Triples (x1014)	Search Using Form Hashing
1	2032	445	445	67.3	445
2	1453	230	230	9.25	230
3	2240	760	760	335	760
4	970	340	340	30.1	340

Table 3. Field identification results. Most incorrectly projected ones are still within a +/- 10 pixel error.

Query	# of writing segments	Number correctly projected	Number projected +/- 10 pixel error	Number projected +/- 20 pixel error
1	8	6	2	0
2	10	8	1	1
3	14	12	2	0
4	15	10	2	1
5	4	4	0	0
6	17	13	2	3

sample forms due to inexact insertion into the clipboard. Using the alignment process described in Section 4, the skewed writing on the tablet was projected onto the electronic form to identify nearby field labels.

We now illustrate the results of field projection. Figure 3e shows digitizer tablet output from writing on the sample form of Figure 3a. Using the sequence of projective transforms, the writing of Figure 3e was projected onto the original form of Figure 3a. The resulting image is shown in Figure 3f (please zoom in on the images for better viewing). As can be seen, 8 of the 9 text regions are projected close to their correct field labels. In addition, there is close resemblance between such automatically projected writing with their actual physical appearance as shown in Figure 3b. Since all pose computations were done using minimal features, there is some error in pose computations leading to alignment errors at the edges as seen for the phone number field. With higher number of features for alignment, such pose errors can be reduced. It is interesting to note, however, that there is a built-in tolerance to pose errors in the field extraction problem due to the finite space left on the form for the field entries. As long as the projected text is within the space provided for the content against the field label, the correct label can still be recovered. Such tolerance is generally more for the x than the y coordinate. Even so, a neighborhood search of the field labels may still have to be performed.

To test the performance of field projection, we measured the pixel difference between the projected tablet writing and the corresponding field label. The text projected within (+/-5 pixels) was taken as a correct projection. The handwriting data was collected by filling out a total of 80 sample forms showing varying amounts of skew in the image. Each form had 3-10 regions filled including those near the bottom of the form page invisible to the camera. The results of field identification for the writing tested are shown in Table 3. As can be seen, a large fraction of the tablet text projects within =/- 5 pixels to the original field label. The field identification performance indicated is sufficient for further post-processing using attribute label extraction and online OCR to successfully populate an electronic medical record.

6 Conclusions

In this paper we have described methods for form indexing and field projection to enable rapid paper form to electronic conversion without explicit need of scanning filled forms or manual population of the electronic medical records.

References

[1] D. Jacobs, “The space requirements of indexing under perspective projections,” IEEE Trans. PAMI, 1996, pp.330-333.

[2] Y. Lamdan, J. Schwartz, and H.J. Wolfson. Object recognition by affine-invariant matching in Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, pages 335-344, 1988.

[3] F.D. Tsai. Geometric hashing with line features. Pattern Recognition, 27:377-389, 1994.

[4] Tanveer Fathima Syeda-Mahmood: Locating Indexing Structures in Engineering Drawing Databases Using Location Hashing. CVPR 1999: 1049-1055.

[5] W.E.L. Grimson, “On the sensitivity of geometric hashing,” ICCV 1997.

[6] J.Mao, M. Abayan, K. Mohiuddin, “A model-based form processing subsystem”, ICPR’96, pp.691-695, 1996.

[7] T. Watanabe, Q. Luo, N. Sugie, “Layout recognition of multi-kinds of table-form document, IEEE Trans PAMI, vo.17, no.4, pp.432-445, 1995.

[8] Wacom Graphire II Digitier Tablet and Inking Pen, http://www.wacom.com/graphire/4x5.cfm

[9] Aiptek VGA PenCam, Irvine, CA 92618 www.aiptek.com

[10] A. Pizano, “Extracting Line Features from Images of Business Forms and Tables”, ICPR'92, pp. 399-403, 1992

[11] D.S. Doermann, A. Rosenfeld, “The Processing of Form Documents”, ICDAR'93, pp. 497-501, 1993.

[12] A.K. Chhabra, "Anatomy of a Hand-Filled Form Reader", Proc. IEEE Trans. On Application of Computer Vision, pp. 195-204, 1994

[13] F. Cesarini, M. Gori, S. Marinai, "A System for Data Extraction from Forms of Known Class", ICDAR'95, Montreal, Canada, pp. 1136-1140, 1995

[14] J.X. Yuan, Y.Y. Tang, C. Y. Suen, "Four Directional Adjacency Graphs (FDAG) and Their Application in Locating Fields in Forms", ICDAR'95, Montreal, Canada, pp. 752-755, 1995.

[15] R. Safari, N. Narasimhamurthi, M. Shridhar, "Document Registration Using Projective Geometry", ICDAR'95, Montreal, Canada, pp. 1161-1164, 1995.

[16] "Xerox mobile camera document imaging," http://www.ipvalue.com/technology/docs/Xerox_Mobile_Camera_Imaging_Document_Capture.pdf 2004.

[17] Watanabe, T., Luo, Q., Sugie, N., Structure recognition methods for various types of documents, MVA(6), No. 2-3, 1993, pp. 163-176.

Symmetric Color Ratio in Spiral Architecture

Wenjing Jia, Huaifeng Zhang, Xiangjian He, and Qiang Wu

Faculty of Information Technology, University of Technology, Sydney,
PO Box 123, Broadway, NSW, Sydney, Australia
{wejia, hfzhang, sean, wuq}@it.uts.edu.au

Abstract. Color ratio gradient (CRG) is a robust method used for color image retrieval and object recognition. It has been proven to be illumination-independent and geometry-insensitive when tested on scenery images. However, the color ratio gradient produces unsatisfying matching results when dealing with an object which appears rotated by a certain relative angle in the model and target images. In this paper, we adopt the idea of color ratio gradient and develop a new method called Symmetric Color Ratio (SCR) based on a hexagonal image structure, the Spiral Architecture (SA). We focus on license plate images and our aim is to achieve a higher matching rate between the SCR histogram of the images within same class in order to separate different classes of images. Our experimental results demonstrate that the proposed SCR is robust to changes over view angles.

1 Introduction

Using color histograms as a stable object representation over change in view for object recognition was first explored by Swain and Ballard [1][2] who introduced the *color indexing* technique to efficiently recognize objects by matching their color-space histograms. This method, however, did not address the issue of illumination variation. Funt and Finlayson [3] produced a new measurement based on the ratio of color RGB triples in neighboring area to locate objects. Compared with Swain's way, this method is more robust to illumination variation. Other improved methods include illumination-independent color reflection ratios proposed by Nayar and Bolle [4]. Gevers [5][6] further developed the color ratio gradient (CRG) to make it insensitive to the geometry and position of the object, shadows, illuminations, and other imaging conditions.

However, such color ratio gradients suffer the following limitation. When dealing with an object which appears rotated by a certain angle in the target image with respect to the model image, the color ratio gradient produces unsatisfying matching results.

In this paper, we adopt the idea of color ratio gradient and develop a new method called Symmetric Color Ratio (SCR), which is based on a hexagonal image structure, the *Spiral Architecture* (SA) [7]. By taking use of the higher symmetry of the hexagon, as well as the consistent definition of distance between the central pixels and any of its 6 directly-connected neighbors on hexagon-based image structure [8], a completely symmetric operator has been defined.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 204–213, 2006.

We focus on vehicle license plate images and our aim is to achieve a higher matching rate between the SCR histograms of the images that belong to the same class. We say two license plate images belong to the same class when they have similar foreground and background colors, but they may have quite different contents (characters), size and even view angles. The SCR histogram of license plate images has been created and used as a robust feature to separate different classes of images. Our experimental results demonstrate that the proposed SCR histogram is robust to changes over view angles.

The remaining parts of this paper are organized as follows. The color ratio gradient as a feature for object recognition is firstly summarized in Sect. 2. The Spiral Architecture, on which the proposed SCR is implemented, is briefly introduced in Sect. 3. In Sect. 4, the proposed Symmetric Color Ratio is defined in detail, and the similarity measurement is obtained. In Sect. 5, the proposed algorithm is tested. Conclusions are given in Sect. 6.

2 Color Ratio Gradients

Supposing the sensor response is measured on an infinitesimal surface patch of an inhomogeneous dielectric object and the spectral power distribution of illumination is unknown, the body reflection term at location $\vec{x}$ in dichromatic reflection model with narrow-band filtering can be written as [5][6],

$$C_k(\vec{x}) = G_B(\vec{x}, \vec{n}, \vec{s}) E(\vec{x}, \lambda_k) B(\vec{x}, \lambda_k) \tag{1}$$

where $G_B(\vec{x}, \vec{n}, \vec{s})$ is the geometric term dependent on the surface orientation $\vec{n}$ and illumination direction $\vec{s}$, $E(\vec{x}, \lambda_k)$ is the illumination intensity at light wavelength λ_k, and $B(\vec{x}, \lambda_k)$ is the surface albedo at light wavelength λ_k.

Gevers proposed the following color constant color ratio[5][6],

$$M(C_1^{\vec{x}_1}, C_1^{\vec{x}_2}, C_2^{\vec{x}_1}, C_2^{\vec{x}_2}) = \frac{C_1^{\vec{x}_1} C_2^{\vec{x}_2} - C_1^{\vec{x}_2} C_2^{\vec{x}_1}}{C_1^{\vec{x}_2} C_2^{\vec{x}_1} + C_1^{\vec{x}_1} C_2^{\vec{x}_2}} \quad C_1 \neq C_2 \tag{2}$$

expressing the color ratio between two adjacent image pixels at location $\vec{x}_1$ and $\vec{x}_2$ under two different light wavelengths. It can be seen that $M \in [-1, 1]$.

Note that in an infinitesimal area it may be assumed that $G_B(\vec{x}_1, \vec{n}, \vec{s}) = G_B(\vec{x}_2, \vec{n}, \vec{s})$, $E(\vec{x}_1, \lambda_{C_2}) = E(\vec{x}_2, \lambda_{C_2})$, and $E(\vec{x}_1, \lambda_{C_1}) = E(\vec{x}_2, \lambda_{C_1})$ [5]. By substituting (1) into (2) and factoring out dependencies on object geometry and illumination direction, we have,

$$\begin{aligned} M(C_1^{\vec{x}_1}, C_1^{\vec{x}_2}, C_2^{\vec{x}_1}, C_2^{\vec{x}_2}) &= \frac{C_1^{\vec{x}_1} C_2^{\vec{x}_2} - C_1^{\vec{x}_2} C_2^{\vec{x}_1}}{C_1^{\vec{x}_2} C_2^{\vec{x}_1} + C_1^{\vec{x}_1} C_2^{\vec{x}_2}} \\ &= \frac{B(\vec{x}_1, \lambda_{C_1}) B(\vec{x}_2, \lambda_{C_2}) - B(\vec{x}_2, \lambda_{C_1}) B(\vec{x}_1, \lambda_{C_2})}{B(\vec{x}_2, \lambda_{C_1}) B(\vec{x}_1, \lambda_{C_2}) + B(\vec{x}_1, \lambda_{C_1}) B(\vec{x}_2, \lambda_{C_2})} \end{aligned} \tag{3}$$

It is seen that color ratio is independent of light intensity, color, viewing condition, and object geometry characteristic. It is determined by the ratio of surface albedo only [5].

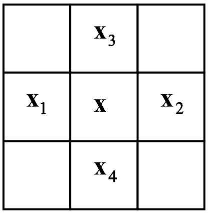


Fig. 1. Locations of four neighbors that are involved in the computation of the color ratio gradient at the central pixel $\vec{x}$

Gevers also defined the gradient of the color constant color ratio as [5][6],

$$\nabla M(C_1^{\vec{x}_1}, C_1^{\vec{x}_2}, C_2^{\vec{x}_1}, C_2^{\vec{x}_2}) = \left(M\left(C_1^{(x-1,y)}, C_1^{(x+1,y)}, C_2^{(x-1,y)}, C_2^{(x+1,y)}\right)^2 + M\left(C_1^{(x,y-1)}, C_1^{(x,y+1)}, C_2^{(x,y-1)}, C_2^{(x,y+1)}\right)^2 \right)^{\frac{1}{2}} \tag{4}$$

where $(x-1, y)$, $(x+1, y)$, $(x, y-1)$, and $(x, y+1)$ are locations of four adjacent neighbors of $\vec{x} = (x, y)$, as shown in Fig.1.

On standard RGB color space, the three-channel color ratios can be written as,

$$\begin{cases} M(R^{\vec{x}_1}, R^{\vec{x}_2}, G^{\vec{x}_2}, G^{\vec{x}_1}) = \dfrac{R^{\vec{x}_1}G^{\vec{x}_2} - R^{\vec{x}_2}G^{\vec{x}_1}}{R^{\vec{x}_2}G^{\vec{x}_1} + R^{\vec{x}_1}G^{\vec{x}_2}} \\ M(R^{\vec{x}_1}, R^{\vec{x}_2}, B^{\vec{x}_2}, B^{\vec{x}_1}) = \dfrac{R^{\vec{x}_1}B^{\vec{x}_2} - R^{\vec{x}_2}B^{\vec{x}_1}}{R^{\vec{x}_2}B^{\vec{x}_1} + R^{\vec{x}_1}B^{\vec{x}_2}} \\ M(G^{\vec{x}_1}, G^{\vec{x}_2}, B^{\vec{x}_2}, B^{\vec{x}_1}) = \dfrac{G^{\vec{x}_1}B^{\vec{x}_2} - G^{\vec{x}_2}B^{\vec{x}_1}}{G^{\vec{x}_2}B^{\vec{x}_1} + G^{\vec{x}_1}B^{\vec{x}_2}} \end{cases} \tag{5}$$

Thus, by substituting (5) into (4), three-channel color ratio gradients on RGB space can be easily obtained. Each $\nabla M(C_1^{\vec{x}_1}, C_1^{\vec{x}_2}, C_2^{\vec{x}_1}, C_2^{\vec{x}_2})$ in (4) can be viewed as being computed in a Quasi-Prewitt operator, as illustrated in Fig.1, where $\vec{x}$ is the central (current) pixel. It can be seen that respectively two neighbors in two directions, i.e., horizontal and vertical directions, have been involved in the computation of the color ratio gradient in the central pixel located at $\vec{x}$.

3 Spiral Architecture

Color ratio gradient is defined on the conventional square-based image structure. Hexagon-based image structure, however, due to its higher symmetry and consistent distance definition between any two adjacent neighbors, can simplify the algorithm design and has attracted many people to do research on it for more than 40 years [8]. In this project, we take use of the above advantages of the hexagon-based image structure and propose the symmetric color ratio (SCR). Our SCR is implemented based on a relatively new hexagon image structure, called *Spiral Architecture* (SA).

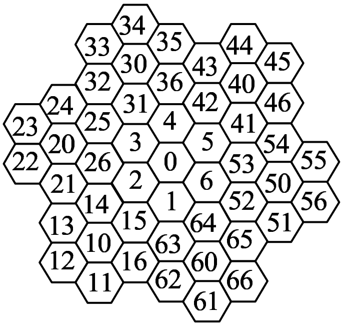
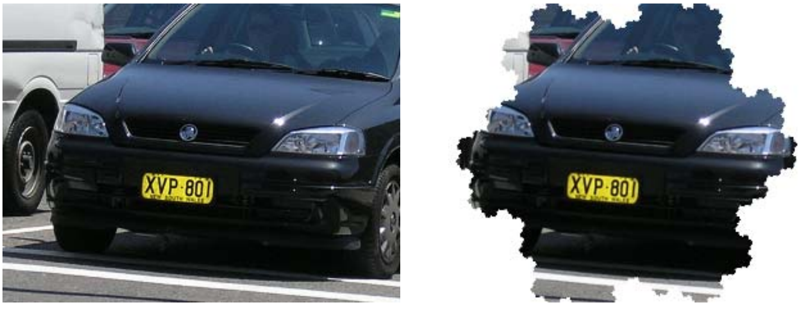


Fig. 2. The Spiral Addressing

The Spiral Architecture (SA) is a unique image representation scheme, proposed by Sheridan [7]. It represents a digital image as a collection of hexagonal pixels, where each hexagonal pixel is addressed in power of seven with a pattern of spiral (see Fig. 2). The hexagon-based image representation and the unique spiral addressing scheme, together with two later proposed mathematic operations, *Spiral Addition* and *Spiral Multiplication*, is called *Spiral Architecture* (SA) [7]. For more details, please refer [7].

Since there is currently no mature hardware device to sample and display images based on hexagonal grids, researchers on hexagonal-based image processing have to use square pixels to mimic hexagonal pixels. Wu et al. [9] constructed a novel mimic scheme called *virtual Spiral Architecture*, on which images on square grids can be smoothly converted to or from virtual Spiral Architecture in order to test algorithms based on hexagon grids. However, this mimic scheme unavoidably introduces certain loss of resolution of image information which results in blur effects. Fig.3 gives a pairs of vehicle images which are represented on normal square structure and virtual Spiral Architecture respectively. Imaging area in Spiral Architecture, as shown in Fig.3(b), contains $7^6 = 117649$ hexagonal grids. In order to avoid the blur effects of approximation on the comparison between the experimental results obtained on two different image structures,


(a)

(b)

Fig. 3. (a) Vehicle image and (b) its representation on Spiral Architecture (SA) experiments on square structure are implemented on the virtual SA-processed images to make the results comparable. Also, the number of pixels in the images is calculated as the total number of valid pixels.

4 Symmetric Color Ratio

In color ratio gradient (CRG) algorithm, only color ratios between two pixels in horizontal and vertical directions (see Fig.1) are involved into the computation of the final color ratio gradient. Once the objects have rotated with a certain angle relative to the background, however, the color ratio gradient at each pixel will be changed accordingly, and hence the CRG histogram is sensitive to the rotation.

In our method, a symmetric color ratio will be proposed. The contribution of doing so is that a symmetric definition takes into account the contribution of the color changes along three diagonal directions rather than horizontal and vertical directions only. In a local area with a cluster of seven pixels, for example, three directions are able to describe the color changes near the central pixel adequately. Although the similar idea may also possibly be applied in square grids, the inconsistent definition of the distance between the two neighbor pixels in diagonal direction and horizontal/vertical directions in square grids always brings uncertainty about the contribution of each component to the combined value. Hexagon-based algorithm, thanks to its symmetric structure, does not meet such kind of trouble. In our algorithm, the color ratio gradient is computed in a window as shown in Fig.4. The window size should not be chosen too large. Otherwise, it breaches the assumption in equation (3), i.e., calculation must be performed on an infinitesimal surface area. We explain the algorithm in detail as follows.

4.1 Symmetric Color Ratio in Spiral Architecture

Let $M_{\vec{x}}$ denote the color ratio between two horizontal neighbors of a pixel and $M_{\vec{y}}$ denote the color ratio between two vertical neighbors of a pixel, where the

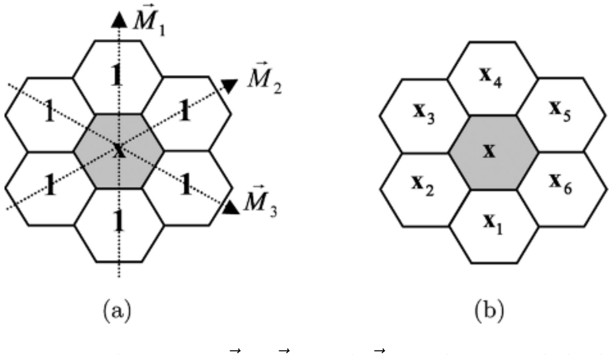


Fig. 4. (a) Symmetric color ratios $\vec{M}_1$, $\vec{M}_2$, and $\vec{M}_3$ at the central shadowed hexagonal pixel $\vec{x}$ in 3 directions. (b)The locations of the six directly connected neighbors of the central hexagonal pixel in Spiral Architecture (SA).

subscript $\vec{x}$ denotes horizontal direction and $\vec{x}$ denotes the vertical direction. In the conventional square grids, the definition of the color ratio gradient in (4) can be simplified as:

$$\nabla M = \sqrt{M_{\vec{x}}^2 + M_{\vec{y}}^2} \tag{6}$$

Following the same naming convention, we use ∇M_{SCR} to denote the symmetric color ratio of an image at a given reference hexagon point $\vec{x}$. Without loss of generality, we define M_1, M_2, and M_3, as shown in Fig.4(a), to denote three color ratios in three diagonal directions respectively as:

$$\begin{cases} M_1(C_i^{\vec{x}_1}, C_i^{\vec{x}_4}, C_j^{\vec{x}_1}, C_j^{\vec{x}_4}) = \dfrac{C_i^{\vec{x}_1} C_j^{\vec{x}_4} - C_i^{\vec{x}_4} C_j^{\vec{x}_1}}{C_i^{\vec{x}_4} C_j^{\vec{x}_1} + C_i^{\vec{x}_1} C_j^{\vec{x}_4}} \\ M_2(C_i^{\vec{x}_2}, C_j^{\vec{x}_5}, C_i^{\vec{x}_2}, C_j^{\vec{x}_5}) = \dfrac{C_i^{\vec{x}_2} C_j^{\vec{x}_5} - C_i^{\vec{x}_5} C_j^{\vec{x}_2}}{C_i^{\vec{x}_5} C_j^{\vec{x}_2} + C_i^{\vec{x}_2} C_j^{\vec{x}_5}} & i \neq j \\ M_3(C_i^{\vec{x}_3}, C_j^{\vec{x}_6}, C_i^{\vec{x}_3}, C_j^{\vec{x}_6}) = \dfrac{C_i^{\vec{x}_3} C_j^{\vec{x}_6} - C_i^{\vec{x}_6} C_j^{\vec{x}_3}}{C_i^{\vec{x}_6} C_j^{\vec{x}_3} + C_i^{\vec{x}_3} C_j^{\vec{x}_6}} \end{cases} \tag{7}$$

where $\{\vec{x}_i\}_{i=1,2,\cdots,6}$ are the locations of six neighbors of the point $\vec{x}$. The locations of six neighbors of a current (central) pixel are illustrated in Fig.4(b).

Thus we have the Symmetric Color Ratio (SCR) defined as:

$$\nabla M_{SCR} = \sqrt{{M_1}^2 + {M_2}^2 + {M_3}^2} \tag{8}$$

In Spiral Architecture, since the distance between the central point and any of its six neighboring points is identical, the resulted symmetric color ratio ∇M_{SCR} is symmetric in three directions rather than two directions, and thus less sensitive to the rotation.

4.2 Similarity Measure

A similarity function is needed to return a numerical measure of similarity between the model and target images. In this paper, we use the histogram of the proposed symmetric color ratio (SCR) as a feature of the model and target images in order to numerically measure the similarity between each other. The advantage of using histogram is the robustness to geometric changes of projected objects.

A three-dimensional SCR histogram $\vec{H}$ is created and chosen as the measurement to compare the similarity between the SCR histograms of two images. The three axis of SCR histogram $\vec{H}$ represent values of SCR between R and G components, simplified as ∇M_{RG}, between R and B components, simplified as ∇M_{RB}, and between G and B components, simplified as ∇M_{GB} respectively. The value $h(i, j, k)$ of each unit in the histogram $\vec{H}$ denotes the total number of frequencies of which ∇M_{RG}, ∇M_{RB}, and ∇M_{GB} take values of i, j, and k respectively.

In order to make such matching invariant to the dimension of image, the created histogram is normalized by the total number of pixels in the image. By such a way, the object matching problem is converted to a simple problem that, to what extent the SCR histogram created for the model image is like the SCR histogram created for the target image.

One straightforward method to calculate the matching rate between two histograms is histogram intersection [2].

Assume the SCR histograms of the model image and target image are $\vec{H}_{Mdl}$ and $\vec{H}_{Tgt}$ respectively, the histogram intersection between the pair of histograms can be defined as:

$$\vec{H}_{Mdl} \cap \vec{H}_{Tgt} = \frac{\sum_{i,j,k=1}^{n} min(h_{Mdl}(i,j,k), h_{Tgt}(i,j,k))}{\sum_{i,j,k=1}^{n} h_{Mdl}(i,j,k)} \tag{9}$$

where n denotes the dimension (bin size) of each axis. It can be seen from (7) and (8) that $\nabla M_{SCR} \in [0, \sqrt{3}]$. In this paper, we take identical bin sizes $n = 100$ for three axis, i.e., ∇M_{SCR} are normalized into the range of $[0, 100]$ for the convenience of computation.

When both $\vec{H}_{Mdl}$ and $\vec{H}_{Tgt}$ are normalized properly, i.e., $\sum_{i,j,k=1}^{n} h(i,j,k) = 1$, (9) can be simplified as:

$$\vec{H}_{Mdl} \cap \vec{H}_{Tgt} = \sum_{i,j,k=1}^{n} min(h_{Mdl}(i,j,k), h_{Tgt}(i,j,k)) \tag{10}$$

A higher histogram matching rate indicates that a better matching between the SCR histograms of model and target images. Higher matching rates are expected when the model image and the target images are within the same class, or when they have similar foreground and background colors, but may have quite different content, size and even view angle.

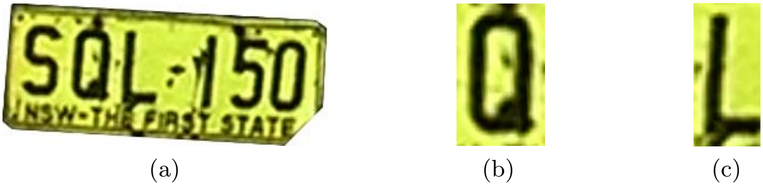
5 Experimental Results

In this project, we focus on license plate images and our aim is to achieve a higher matching rate between the SCR histograms of two license plate images that are taken from the same class.

The experiments are finished in three parts. In Sect. 5.1, the performance of the proposed SCR histogram is evaluated where SCR is taken as a feature to separate license plate images that belong to different classes. In Sect. 5.2, the independence of SCR on model images which contain quite different characters is proved. In Sect. 5.3, the SCR histogram is shown insensitive to the model images which appear to have a rotation angle. The details are explained as follows.

5.1 Similarity Measurement

Matching rates between the model image and the target images within same class are expected to be relatively high, while they should be very low between different classes of images.



(a) (b) (c)

Fig. 5. An example of cutting a small part of a license plate image(a) to form model images(b)(c)

In this experiment, without loss of generality, two classes of vehicle license plate images are tested, namely, license plates with yellow background and license plates with white background. We say two license plate images belong to the same class when they have similar foreground and background colors, but they may have quite different content (characters), size and viewing conditions. For each class, we randomly select a license plate image and cut a small part of image from it (see for example Fig.5.) as the model image of this class. Then, the SCR histogram of the model image is computed and matched with the SCR histograms of the various license plate images (target image) within same class and from another class.

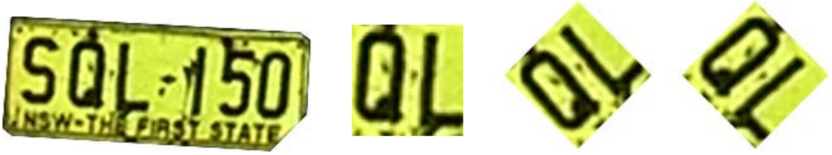
The selection of the model image is quasi-random. Any part of the license plates that contains at least one complete character can be chosen as a model image. Obviously, the larger the size of the model image is, the longer processing time will be needed.

The experiment is done on 64 yellow plates and 27 white plates with different characters, sizes, orientations and illumination conditions. According to our experiments, an average matching rate of 83.5% within the same class can be obtained, while the average matching rate is 44.4% for two images that are taken from different classes. This demonstrates that the SCR histogram can be used as robust feature to separate the different classes of license plate images easily.

5.2 Insensitiveness to the Content of Model Images

License plates in Australia may contain characters including twenty-six capital letters $A \sim Z$ and ten Arabic digits $0 \sim 9$. As we mentioned in previous subsection, the model image should be selected quasi-randomly. This is to say, no matter which character has been included in the model image, the similarity measurement, i.e., the similarity between the model image and the target images, should be very high and stable.

In this experiment, the model image is still chosen from cutting at least one complete character from the license plate images. However, we cut different parts of a license plate image which contain different characters as model image to test the matching rate between their SCR histograms. The different characters that we chose have large appearance difference, such as "Q" and "L", of which the former contains more curve edge information, while the latter contains more linear edge information.



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

Fig. 6. Experiments-3: Similarity measurement with rotated model images: an example. (a) Target image; (b) model image without rotation relative to (a); (c) model image with 45^o rotation relative to (a); (d) model image with -45^o rotation relative to (a).

The experiments give stable matching rates for most character structures. For example, the matching rates of SCR histogram for the case shown in Fig. 5 using model images that contain "Q" and "L" are 81.6% and 81.3% respectively. This explains that the matching result using SCR histogram is insensitive to the content inside the images.

5.3 Insensitiveness to Object Rotation

In automatic license plate recognition practice, the vehicle images may be tilted to some extent due to uneven or curvy road surface. As a result, the license plate on vehicle may appear to be rotated with a certain angle to the background, which results that the vertical and horizontal gradient information along edges of characters in license plates will be changed to some extent. However, when SCR is used, the histogram matching rate should be stable due to the symmetric feature of the algorithm.

In this experiment, we keep the characters of model image and the target images unchanged, but rotate the model image with a certain angle relative to the model image, as shown in Fig. 6, then compute the SCR histogram matching rate respectively and compare them.

The matching rates of SCR histogram between the rotated model image and the target images from the same class remain significantly higher than the matching rates obtained for images from different classes. Moreover, for some character structures, the SCR histogram gives more stable matching rate than CRG histogram. For example, the matching rate between the target image in Fig. 6(a) and model images in Fig. 6(b), (c), and (d) are 87.5%, 88.4% and 88.3% respectively when using SCR histogram for histogram matching. While using CRG histogram, these matching rates are 86.5%, 76.2% and 76.4% respectively. This demonstrates hat the SCR histogram is less sensitive to the rotation of the objects and further proves the robustness of the proposed SCR histogram.

6 Conclusions

The color ratio gradient algorithm has limitations while applied to objects that appear relatively rotated with a certain angle with respect to the background. In this paper, we adopt the idea of color ratio gradient and develop a Symmetric Color Ratio (SCR), which considers three directions around a central pixel with identical symmetric feature. The SCR histogram is applied to license plate images in order to find a higher similarity measurement between images that belong to same class. Our experimental results show that the SCR histogram is insensitive to different characters, sizes, colors, orientations and illumination conditions when being applied to separate license plate images. Besides, the independence of the algorithm on the model images and the robustness to the changes over rotation angle has also been proven. This demonstrates that the proposed algorithm can be used as a robust feature for license plate images.

Acknowledgement

The work in this paper is supported by Australian Research Council (ARC) Large Research Grant No. DP0451666.

References

1. Swain, M.J. and D.H. Ballard, "Indexing via color histograms", Computer Vision 1990. Proceedings, Third International Conference on, 1990: p. 390-393.
2. Swain, M.J. and D.H. Ballard, "Color Indexing", International Journal of Computer Vision, 1991. 7(1): p. 11-32.
3. Funt, B.V. and G.D. Finlayson, "Color constant color indexing", Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1995. 17(5): p. 522-529.
4. Nayar, S.K. and R.M. Bolle, "Reflectance ratio: A photometric invariant for object recognition", Computer Vision, 1993. Proceedings, Fourth International Conference on, 1993: p. 280-285.
5. Gevers, T. and W.M. Smeulders, "Color constant ratio gradients for image segmentation and similarity of texture objects", Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001. 1: p. I-18-I-25 vol.1.
6. Gevers, T., "Image segmentation and similarity of color-texture objects", Multimedia, IEEE Transactions on, 2002. 4(4): p. 509-516.
7. P. Sheridan, "Spiral Architecture for machine vision," PhD Thesis, University of Technology, Sydney, 1996.
8. X. He and W. Jia, "Hexagonal Structure for Intelligent Vision", Proceedings of the 1st International Conference on Information and Communication Technology (ICICT 2005) (IEEE), Karachi, Pakistan, August 2005.
9. Q. Wu, X. He, and T. Hintz, "Virtual Spiral Architecture," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, vol. 1, pp. 399-405, 2004.

A Geometric Contour Framework with Vector Field Support

Zhenglong Li, Qingshan Liu, and Hanqing Lu

National Laboratory of Pattern Recognition,
Automation of Institute, Chinese Academy of Sciences,
P.O. Box 2728, Beijing, Zip Code: 100080
{zlli, qsliu, luhq}@nlpr.ia.ac.cn

Abstract. In this paper, we propose a new geometric contour framework with support of specified vector field. First we define three criteria for selection of vector field in geometric model. According to the criteria, EdgeFlow, a powerful segmentation tool, is selected to generate desirable initial vector field. In order to overcome the drawbacks of conventional geometric models, multi-source external forces, such as from texture and multi-spectra, are integrated to provide the ability for segmenting the texture-rich and complex scene images. Instead of common smoothing pre-processing to denoise and suppress possible spurious edges, the more advanced complex diffusion filters are adopted in our algorithm, which result in the piecewise filtered image to help detect those sharp transition regions. We test our model on the Berkeley Segmentation Database, and the experimental results are promising.

1 Introduction

In recent decades, *Curve Evolution* methods have been extensively exploited in computer vision society due to their potential applications in object contour extraction/object segmentation, motion estimation&tracking, et al. According to the expressions of mathematical models, they are roughly divided into two classes: *Parametric Curve Model* and *Geometric Curve Model/Non-Parametric Curve Model*. The representative of the parametric curve models is *Snakes* [1]: it drives a parameterized curve by image-oriented force to positions of interests. And the geometric curve model is based on the theory of *Level Set* [2] which can deal with topological changes during evolution process without any additional intervention. Contrast with the constraints of snakes, geometric curve models have two advantages: 1) They can deal with the case that the number of objects in scene is unknown; 2) No additional re-parameterization maneuver is needed during curve evolution.

The original geometric curve models were introduced by Malladi et al [3] and Caselles et al [4] respectively. The ideas of them utilize the grayscale/single spectral information as the external force to drive curve to positions of interests. Though these algorithms achieve successes to some extent, there also exist some limitations. First, most them use simple grayscale gradient, so that they merely

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 214–223, 2006.

tackle “simple” images, i.e. synthesized images or images with clear edge cues. As for the rich textural context or presence of clutters, they fail to converge to positions of interests. Second, usually only single source image-oriented information is integrated into model, whereas other useful and important information such as color, texture do not get their places in curve evolution. For those models using single cue as driven force, it is difficult to get reliable external force in those region of indistinct single cue. Some researchers [5, 6] have noted these situations and proposed solutions for textural images. Recently Xie [7] gives an interesting attempt to integrate information of region-based segmentation into model.

In this paper, we propose a novel curve evolution framework, which can handle segmentation in some relatively texture-rich and/or complex context images. We first define three criteria for customized vector field. Based on these criteria, an effective algorithm, i.e., *EdgeFlow* [8], is selected to generate desirable initial vector field. In order to deal with the drawbacks of the conventional geometric models, multi-source external forces, such as forces from texture and multi-spectra, are integrated to provide the ability to segment the texture-rich and complex scene images. We adopt a so-called “generalized gradient” method to extract change information in multi-spectral/vector images (we treat color images and Gabor filtered multi-channel images as multi-spectral image in a sense of signal processing). “Generalized gradient” has been proven an effective and simple method in multi-spectral image analysis [9]. Through proper diffuse processing of initial vector field, the desirable vector fields are achieved. Instead of common smoothing pre-processing to denoise and suppress possible spurious edges, the more advanced complex diffusion filters are adopted in our algorithm, which result in the piecewise filtered image to help detect those sharp transition regions in image. We test our model on the Berkeley Segmentation Database and the experimental results show effectiveness of our method.

The proposed method is different from [6]. The difference is that they try to divide vector field of EdgeFlow into the irrational and solenoidal vector fields. By solving a Poisson PDE, they get an edge function V, and then edge flow and edge function are integrated into geometric contour framework. But this method directly relies on pre-processing of EdgeFlow. Instead of depending on any single existing vector field generation scheme, we propose the general criteria for customizing vector field in curve evolution. Any vector field that satisfies those criteria can be taken into the geometric curve model. Moreover, multi-source cues can be integrated into the model to get reliable result. There also exist the differences between our method and [7]. In [7], they use pre-segmentation by Mean Shift to derive region-related driven force to overcome edge leakage and then GVF to extend capture range. And for pre-segmentation map, direct GVF diffuse process cannot guarantee a regular vector field under clutter pre-segmentation result or complex scene. This method still only segment simple images, and fails in dealing with textural or complex scene images. The proposed framework combines diffusion enhancement and vector field generation method to overcome these disadvantages.

2 A Brief Review of Geometric Contour Model

Geometric active contour model is based on the curve evolution theory. Usually it needs initial curve C_o to be specified beforehand, and it uses the external force derived from image to drive curve to positions of interests. The process of curve evolution is described by the coupled partial differential equations (PDEs). Compared with conventional parametric numerical methods, Level Set method [2] has the advantages of adaption to topological changes, i.e. curves merging and/or splitting in an automatic manner without any additional intervention, while parametric contour model cannot easily handle those situations. Some works based on parametric contour model try to deal with topological changes, but they bring with a huge computation complexity [10].

The geometric contour model is introduced respectively by Malladi et al [3] and Caselles et al [4]. In [3], the following PDE is used as

$$C_t = g(|\nabla I|)(\kappa + c)\vec{N}, \tag{1}$$

where C is a 2-D closed contour, $g(\cdot) : [0, \infty] \mapsto [1, 0]$ is a monotonic decreasing function, κ and c are curvature and constant speed item respectively, and $\vec{N}$ is the unit normal vector of curve with inward direction. It is obvious that $g(\cdot)$ will decrease quickly to zero when contour is near edge, so the advancing curve will be stalled for total speed C_t approximating to zero. In (1) constant c is an artificial balloon-like force [11] to drive curve in those feature-lacking region to contract/expand to object boundary. However this model has a defect that curve cannot come back if the curve steps beyond object boundary.

In [4], an improved model is proposed as follows

$$C_t = g(|\nabla I|)(\kappa + c)\vec{N} - (\nabla g(|\nabla I|)) \cdot \vec{N})\vec{N}, \tag{2}$$

in which an additional item is introduced in right hand side of (2) compared with (1). This new item offers curve the ability to come back when going beyond object boundary. But the capture range for pure gradient-deduced vector force, $(\nabla g(|I|)) \cdot \vec{N})\vec{N}$, is very limited, because it is determined by used gradient operator, e.g. for the most used derivative of Gaussian operator, capture range is affected by σ.

Because stopping items in (1) and (2) is too weak to counter geometric force to get balance, in the case that image-derived gradient is small, both model (1) and (2) fails to handle weak edges.

3 Our Works

3.1 Criteria for Vector Field

From a viewpoint of dynamics, there are two kinds of strategies to stop the motion of the curve. One is considering scalar function $g(\cdot)$ as edge indicator such that $g(\cdot)$ will attenuate to zero quickly, and therefore decrease the total speed in (1) and first speed item in (2) to zero, when encountering high change rate of local feature, e.g. gradient as the most usual choice. The other is to add vector force $\nabla g(\cdot)$ to balance the image-oriented force at positions where vector flows bump into. The convergence is achieved only when whole system gets to equilibrium between the geometric constraint and the image-oriented forces. Both strategies employed in (1) and in (2) cannot handle weak edges for noise, illumination, and albedo et al. Only relying on $g(\cdot)$ or gradient flow from gradient operator is not enough to stop curve moving over those weak edges and/or scarce feature regions.

In this paper, we design a novel framework to deal with problems that conventional models cannot solve, such as edge leakage and applications in texture-rich and/or complex scene. Instead of designing complicated $g(\cdot)$ or considering to design new external force, we consider that the drawbacks, i.e. weak edge leakage, capture range, and textural image segmentation, et al, can be rescued by a vector flow field which satisfies some criteria, from a viewpoint of separation of underlying vector field from the geometric contour model. In the following, we give the definition of these criteria.

Many studies have contributed to the solution of the problems in curve evolution. In [11], Cohen et al mentioned only using direction information of gradient vector as external driven force by which head-to-head vector flows between edge are constructed, and this scheme can make contour converge to object boundary. The idea of Xu et al [12] and Yuan [13] are similar.They all deduce vector field from edge map, for those vectors in the regions far from edges, and all can be traced back to edges. From the viewpoint of field theory, we can regard those methods as generating the conservative field, in which every point of curve acts like a free particle effected by the conservative field force and geometric constraints.

Here, we propose three new criteria for vector field in geometric model. 1)*Direction Criterion.* Between edges, vector flow should be head-to-head pointing toward each other; 2) *Energy Criterion.* Vectors in the vicinity of edges should possess dominant energy; 3) *Attraction Criterion.* For any region far from edges, there should exist vector flow pointing toward edge or the vector tracing back toward edge. We give some brief explanations on these three criteria as follows.

The purpose of criterion 1) is to stop moving curve and accurately locating edge positions. Assuming some part of deforming curve is across the edge while countering constant balloon-like force and geometric constraint, it is possible that parts of curve moves beyond edges. Thus, we require the head-to-head vector flow to draw curve back when curve is beyond edge location. Moreover, if we remove balloon-like force to allow curve under effect of only vector flow and geometric constraint, we will get an improvement of final convergence to get accurate positioning to edge. But it brings with a problem that contour model will become sensitive to noise points and/or meaningless blobs. The *energy criterion* we define is of two-fold purposes. First, around edge the attraction should reach extrema to trap deforming contour. Second, dominant energy can accelerate convergence process near edges, contrast with the method of only

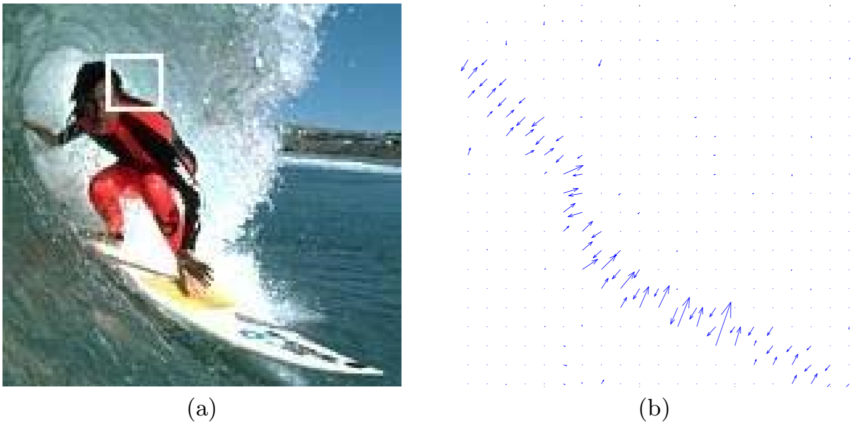


Fig. 1. Vector field formed by EdgeFlow method. (a) The part of original image. (b) Magnified vector field by EdgeFlow in white square region of (a).

using directional information [11]. Since initial contour may be not prescribed exactly in vicinity of object. In those regions far from edge, the deforming curve is wished to be still attracted by edge, while balloon-like force should be set as small as possible for the consideration of edge leakage. That's the reason why we contrive *attraction criteria.*

One aim of our framework is that any vector field satisfying these three criteria above can be integrated into the geometric model. To demonstrate the effectiveness of the criteria, we choose EdgeFlow [8] as the candidate vector field, although edge flow does not fully satisfy the criteria we define above. But we find that if we diffuse edge flow vectors with GVF method [12], the modified vector field can satisfy well all the three criteria.

EdgeFlow is a powerful tool for boundary detection and image segmentation. It cannot only incorporate single grayscale information, but also other information such as the color, Gabor phase and et al (for details, cf. [8]). This method was tested on a large number of natural images, and gave the good performance. Fig. 1(b) shows the edge flows in the white square of the left subfigure (a).

3.2 Complex Diffusion and Generalized Gradient

In order to suppress spurious edges and reduce effect of noise, smoothing operation is an usual pre-processing method before any operation extracting useful information in images. The widely used multi-scale smoothing scheme is Gaussian pyramid. But this scheme will bring with the implementation difficulties for curve evolution across different scales [14] and lose accurate edge locating ability. To deal with these problems, a more advanced diffusion process called complex diffusion process is adopted in this paper. The complex diffusion process produces filtered images with piecewise properties.

For multi-spectral/multi-value images, simply algebraic combination of each channel's response cannot give a fine description on change rate of local features.

We use “generalized gradient” [15] to describe local feature changes in multi-spectral images, for it is a robust and effective method ever proven.

Complex Diffusion. In low level vision tasks, filtering is often used to reduce the influence of noise and meaningless clutter. For the images with rich texture or complex scene, conventional filtering cannot fulfil this task. Scale-space approach is a proven useful technique in image processing. And it is known that Perona-Mallik diffusion is an adaptive diffusion process to the different scale, although this scheme cannot deal with texture-rich image for its sensibility to variation of gradient. Shock filters [16] suffer the same shortcoming. We use the complex diffusion [17] to do smoothing inside homogeneous region while enhancing edge. We use this more advanced PDE-based diffusion process to smooth inside region while sharpening edge. The complex shock filter used in this paper is

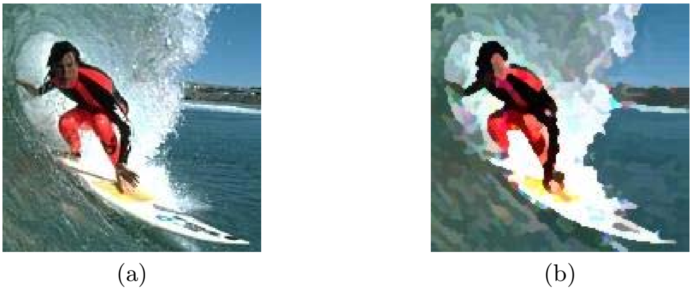
$$I_t = -\frac{2}{\pi} \arctan\left(a\,\mathrm{Im}(\frac{I}{\theta}) \right) |\nabla I| + \lambda I_{\eta\eta} + \tilde{\lambda} I_{\xi\xi}, \tag{3}$$

where $\lambda = re^{i\theta}$, and $\tilde{\lambda}$ is a real scalar value (for more details, cf. [17,16]).

The complex diffused image is composed of real part and imaginary part, i.e. the value of each point is a complex value. The real part and imaginary part of complex filters play the different roles in diffusion process. The real part of complex filters behaves like an adaptive smoothing operator, while the imaginary part of complex filters is similar to a simple second derivative operator, but offers more advantages than it.

For multi-spectral images, e.g. color images and Gabor-filtered images, we apply this complex shock filter to each sub-band image respectively to denoise and enhance its edges, and then “generalized gradient” will be extracted from sub-band images to represent the change rate cross all spectra. Fig. 2 gives a complex diffusion processed image. It is obvious to see that edges are enhanced and inside region is well smoothed. In fact, this diffusion process can be seen as a reconstruction process of piecewise image.

Generalized Gradient. For a multi-spectral/multi-value image I with m channels, we can regard the imaging process as a function $f : \mathbb{R}^2 \mapsto \mathbb{R}^m$. A point in



(a) (b)

Fig. 2. Result by complex shock filter. (a) The original image. (b) The filtered image with piecewise property.

the image is a m-vector. Now, we use the approach named "generalized gradient" to detect changes in local feature of $\mathbb{R}^m$.

Consider two points P and Q in $\mathbb{R}^m$, i.e. two m-vectors respectively. Difference between them is $\Delta\mathbf{f} = \mathbf{f}(P) - \mathbf{f}(Q)$. When the distance between P and Q is infinitesimal, i.e. $d(P, Q)$ tending to zero, we can get its squared norm in matrix notation [15]

$$\mathrm{d}\mathbf{f}^2 = \begin{bmatrix} \mathrm{d}x_1 \\ \mathrm{d}x_2 \end{bmatrix}^T \begin{bmatrix} g_{11} \; g_{12} \\ g_{21} \; g_{22} \end{bmatrix} \begin{bmatrix} \mathrm{d}x_1 \\ \mathrm{d}x_2 \end{bmatrix}, \tag{4}$$

where

$$g_{ij} = \sum_{k=1}^{m} \frac{\partial f_k}{\partial x_i} \frac{\partial f_k}{\partial x_j}.$$

Note that $d\mathbf{f}^2$ indicates the change rate along direction $P - Q$. In the directions of two eigenvector of matrix $[g_{ij}]$, the equation (4) gets two extrema, i.e. the maximum value and the minimum value respectively. Here $[g_{ij}]$ is a 4×4 2-D matrix, and its eigenvalues are given as follows

$$\lambda_{\pm} = \frac{g_{11} + g_{22} \pm \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2}. \tag{5}$$

and the corresponding eigenvectors are

$$l_{\pm} = (\cos\theta_{\pm}, \sin\theta_{\pm})^T, \tag{6}$$

where

$$\begin{aligned} \theta_{+} &= \frac{1}{2} \arctan \frac{2g_{12}}{g_{11} - g_{22}} + k\pi, \quad \text{and} \\ \theta_{-} &= \theta_{+} \pm \frac{\pi}{2}. \end{aligned}$$

When m, the number of sub-bands, is equal to 1, λ is equivalent to gradient in the case of 2-D single-value image. Thus, it is called "generalized gradient".

Replacing ∇I of gradient with λ_{+}, we integrate components of multi-spectral image information into the geometric model. The change rate in any multi-spectral image can be described by this method. In our experiments, only Gabor-filtered images and color RGB images are used.

In order to modify vector field generated by EdgeFlow for the proposed criteria, we use GVF [12] to diffuse vector field of edge flow. The idea of GVF is to diffuse edge map to get vector field for extending capture range. The GVF's diffusion equations are two coupled PDEs:

$$\mu\nabla^2 u - (u - I_x)(I_x^2 + I_y^2) = 0, \tag{7a}$$

$$\mu\nabla^2 v - (v - I_y)(I_x^2 + I_y^2) = 0. \tag{7b}$$

We modify the edge items $(I_x^2 + I_y^2)$ in (7) as λ_+ in sub-band images and set initial condition of (7) as vector field from EdgeFlow:

$$\mu\nabla^2 u - (u - E_x)\lambda_+ = 0, \tag{8a}$$

$$\mu\nabla^2 v - (v - E_y)\lambda_+ = 0, \tag{8b}$$

where E_x, E_y are x, y component of vector in EdgeFlow.

With integration of EdgeFlow, "generalized gradient", and complex diffusion, we give our geometric model with combination of multi-source external forces as

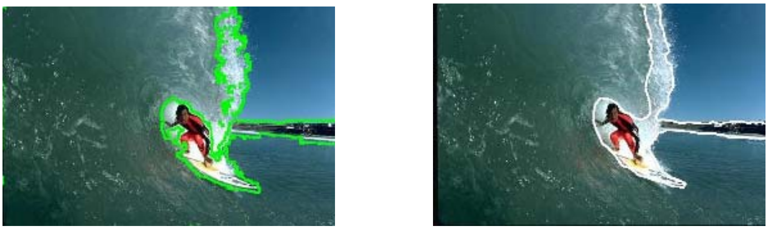
$$C_t(p) = \left(\alpha g(|\nabla I|)(\kappa + c) - \beta\nabla g(\lambda_+)\cdot\vec{N} + \gamma g(|\vec{F}|)\frac{\vec{F}}{|\vec{F}|}\cdot\vec{N}\right)\vec{N}, \tag{9}$$

where $\vec{F} = (u, v)^T$ is the vector field from (8), and α, β, γ are weight coefficients respectively.

In (9), we integrate the multi-source external forces and the customized vector field into our model. The second item $\nabla g(\lambda_+)\cdot\vec{N}$ in right hand side of (9) incorporates the generalized gradient to utilize multi-spectral information. And in the last item $\vec{F}$ is the vector field force derived from edge flow. The customized $\vec{F}$ offers model the ability to segment the texture-rich and/or complex images. Furthermore, with a strategy of multi-source external forces and customized vector field, the proposed model can overcome the drawbacks of conventional models mentioned in previous section.

4 Experiments

We test our model on the Berkeley Segmentation Database. Any vector field which satisfies the proposed criteria can be integrated into (9). In test, we choose EdgeFlow to generate initial vector field (for implementation issues, cf. [8]). Then by (8), we modify the original edge flow vector field to conform to our criteria, so the desirable vector field has been constructed. Moreover, to describe the change



(a) (b)

Fig. 3. Segmentation results by EdgeFlow and our method, respectively. (a) The result of original EdgeFlow. (b) The corresponding result by our method.

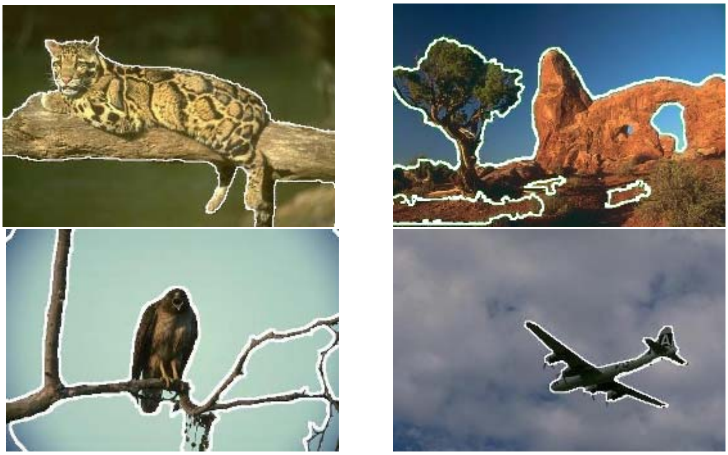


Fig. 4. Some other experimental results by our method

of local feature in multi-spectral image, we use “generalized gradient”. The initial curves are set manually and a post-process of region merging is adopted in the test.

See Fig. 3(a), it is the segmentation result by the original EdgeFlow method [8]. In the middle of the image, we can find the difference between white water wave and surfing man cannot be well segmented by EdgeFlow. And in the right upper part of image, the white water wave has the trend to be over-segmented. Fig. 3(b) shows the corresponding result by our method, and it can be seen that the proposed method gives a finer segmentation result.

Fig. 4 shows some other examples of segmentation with our method. These examples have rich texture and complex scene content. Our method still gives the interesting segmentation results. Although some little defects still exist in segmentation results, it must be pointed out that segmenting image with texture-rich content and/or complex scene is a very challenging work.

5 Conclusion

We propose a novel geometric contour model with support of customized vector field. Multi-source external forces are integrated into model to give high reliable performance. Contrast with conventional model which can only deal with simple image, our model can be applied to relatively complex and textural images. We do tests on Berkeley Segmentation Database and the result is promising.

Acknowledgment

We would like to acknowledge support from Natural Sciences Foundation of China under grant No. 60135020, 60475010 and 60121302.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. Jnl. of Comp. Vis. **1** (1988) 321–331
2. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics **79** (1988)
3. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: A level set approach. IEEE Trans. Pattern Anal. Machine Intell. **17** (1995) 158–175
4. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. Jnl. of Comp. Vis. **22** (1997) 61–79
5. Paragios, N., Deriche, R.: Geodesic active regions and level set methods for supervised texture segmentation. Int. Jnl. of Comp. Vis. **46** (2002) 223–247
6. Sumengen, B., Manjunath, B.S., Kenney, C.: Image segmentation using curve evolution and flow fields. In: IEEE Int. Conf. on Image Processing. (2002)
7. Xie, X., Mirmehdi, M.: RAGS: Region-aided geometric snake. IEEE Trans. Image Processing **13** (2004) 640–652
8. Ma, W., Manjunath, B.S.: EdgeFlow: A technique for boundary detection and image segmentation. IEEE Trans. Image Processing **9** (2000) 1375–1388
9. Cumani, A.: Edge detection in multispectral images. Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing **53** (1991) 40–51
10. McInerney, T., Terzopoulos, D.: Topologically adaptable snakes. In: IEEE Int. Conf. on Comp. Vis., Cambridge, MA (1995) 840–845
11. Cohen, L.D., Cohen, I.: Finite element methods for active contour models and balloons for 2-D and 3-D images. IEEE Trans. Pattern Anal. Machine Intell. **15** (1993) 1131–1147
12. Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. IEEE Trans. Image Processing **7** (1998) 359–369
13. Yuan, D., Lu, S.: Simulated static electric field(SSEF) snakes for deformable models. In: Int. Conf. on Patt. Recog. Volume 1. (2002) 83–86
14. Leroy, B., Herlin, I.L., Cohen, L.D.: Multi-resolution algorithms for active contour models. In: Proc. of the 12th Intl. Conf. on Analysis and Optimization of Systems Images, Wavelets and PDE'S, Rocquencourt, France (1996)
15. Zenzo, S.D.: A note on the gradient of a multi-image. **33** (1986) 116–125
16. Alvarez, L., Lions, P.L., Morel, J.M.: Image selective smoothing and edge detection by nonlinear diffusion. II. SIAM J. Numer. Anal. **29** (1992) 845–866
17. Gilboa, G.: Image enhancement and denoising by complex diffusion processes. IEEE Trans. Pattern Anal. Machine Intell. **26** (2004) 1020–1036

Clustering Spherical Shells by a Mini-Max Information Algorithm

Xulei Yang, Qing Song, Wenbo Zhang, and Zhimin Wang

School of Electrical and Electronics Engineering,
Nanyang Technological University, Singapore 639798
yangxulei@pmail.ntu.edu.sg

Abstract. We focus on spherical shells clustering by a mini-max information (MMI) clustering algorithm based on mini-max optimization of mutual information (MI). The minimization optimization leads to a mass constrained deterministic annealing (DA) approach, which is independent of the choice of the initial data configuration and has the ability to avoid poor local optima. The maximization optimization provides a robust estimation of probability soft margin to phase out outliers. Furthermore, a novel cluster validity criteria is estimated to determine an optimal cluster number of spherical shells for a given set of data. The effectiveness of MMI algorithm for clustering spherical shells is demonstrated by experimental results.

1 Introduction

Clustering plays an important role in many engineering fields such as pattern recognition, system modelling, image processing, communication, data mining, and so on. In most traditional clustering algorithms, the relationship. There is a whole class of fuzzy clustering algorithms in the literature in which an objective function based on a distance measure is iteratively minimized to obtain the final partition. The kinds of partitions they generate or the shapes they detect depend on either the norm, which is used to define the distances, or the kind of cluster prototype used. Fuzzy c-shells (FCS) algorithm [3] utilizes spherical shell as prototype, which has been proved to be successful in detecting the clusters with hollow interiors, especially for circle shapes. The fuzzy c-spherical shells algorithm (FCSS) [5] reduces the computational costs of the FCS by introducing an algebraic (non-Euclidean) distance measure. In this way, the prototypes can be calculated directly and an algorithm to solve coupled nonlinear equations need not be used (as in the FCS). Man and Gath developed the fuzzy c-rings (FCR) [6] for two-dimensional (2D) case, which is computationally much more efficient than the FCS and suffers not from a highly non-Euclidean distance measure like the FCSS.

Although there are special algorithms, as e.g., FCS, FCSS and FCR, to detect and separate spherical shells (ring-shaped data) in the clustering literature, some special issues like the robustness of the algorithms need further improvement. As mentioned in [8], the class of objective function clustering algorithms may suffer from the sensitivity of data initialization. If the cost function used is not

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 224–233, 2006.

convex and has local minima (a typical case), the algorithm may be trapped into one of them, resulting in a non-optimal partition. Obtaining an optimal partition with the cost function converging to global minimum, depends on whether or not a *right* number of cluster prototypes has been assumed at the beginning of the algorithm; and it depends also on whether or not these prototypes have been positioned properly. Some efforts aimed at this non-convex optimization problem have been made successfully. One of the practical algorithms, the deterministic annealing (DA) approach [7] [8], which is independent of the choice of the initial data configuration and has the ability to avoid poor local optima, has demonstrated substantial performance improvement for clustering problem and its extensions over standard supervised and unsupervised learning methods.

In this paper, we focus on the detection and separation of spherical shells by a mini-max information (MMI) clustering algorithm based on mini-max optimization of mutual information (MI), which is basically a two step approach. The clustering phase, i.e., the minimization of the MI, leads to a mass constrained deterministic annealing (DA) algorithm [8], in which the annealing process with its phase transitions leads to a natural hierarchical clustering, resulting in the independence of the choice of the initial data configuration and the ability to avoid poor local optima. The robust pruning phase, i.e., the maximization of MI, provides a robust estimation of probability soft margin to phase out outliers. The trade-off between the optimal clustered data points and the rejection of outliers is controlled through a separated parameter. Furthermore, a novel cluster validity criteria is estimated to determine an optimal cluster number of spherical shells for a given set of data. The subsequent section contains the derivations of MMI algorithm. The experimental results of the proposed MMI method are reported in section 3. Finally, conclusion and discussion are presented in section 4.

2 The Proposed MMI Clustering Algorithm

Assume a given data set $X = \{x_1, x_2, \ldots, x_l\} \subset R^n$ consists of c spherical shells denoted by $W = \{w_1, w_2, \ldots, w_c\}$ with the $k(th)$ prototype w_k consisting of two parameters (v_k, r_k), where $v_k \in R^n$ is the center of the sphere and $r_k \in R$ is the radius. We borrow the distance measure used in FCSS [5] as the distortion measure in our MMI method, i.e., the distortion between x_j and w_k is presented by

$$d(x_j, w_k) = p_k^T M_j p_k + q_j^T p_k + b_j \tag{1}$$

where $b_j = (x_j^T x_j)^2$, $q_j = 2(x_j^T x_j)y_j$, $y_j = [x_j^T \;\; 1]^T$, $M_j = y_j y_j^T$ and $p_k = [-2v_k^T \;\; v_k^T v_k - r_k^2]^T$. Note that all the variables (vectors or matrix) in (1) are only related to the input point x_j except the parameter p_k which is decided by a pair of parameters, i.e. the center v_k and the radius r_k.

In view of the information theory (channel capacity and rate-distortion function [1]), together with the statement in [9] and the deterministic annealing (DA) approach [7], we propose a mini-max information (MMI) clustering algorithm for clustering of spherical shells based on the following mini-max optimization problem:

$$F_{MinMax} = \min_{P(W|X)} \max_{P(X)} \quad (I(X,W) - sD(X,W)) \tag{2}$$

where $D(X,W)$ is the average distortion function defined by

$$D(X,W) = \sum_{j=1}^{l}\sum_{k=1}^{c} p(x_j)p(w_k|x_j)d(x_j,w_k) \tag{3}$$

and $I(X,W)$ is the average mutual information which can be written in either of the two following forms [1]:

$$I(X,W) = \min_{P(W)} \sum_{j=1}^{l}\sum_{k=1}^{c} p(x_j)p(w_k|x_j)\log\frac{p(w_k|x_j)}{p(w_k)} \tag{4}$$

$$I(X,W) = \max_{P(W|X)} \sum_{j=1}^{l}\sum_{k=1}^{c} p(x_j)p(w_k|x_j)\log\frac{p(x_j|w_k)}{p(x_j)} \tag{5}$$

where $p(x_j) \in P(X)$ and $p(w_k|x_j) \in P(W|X)$ denote the source and conditional forward pmf sets respectively, $p(w_k) \in P(W)$ and $p(x_j|w_k) \in P(X|W)$ denote the output and conditional backward pmf sets respectively. The parameter s in (2) is the Lagrange multiplier, which will be substituted by the temperature parameter T in the minimization optimization and the control parameter λ in the maximization optimization respectively. We will show below that the minimization of (2) against the conditional pmf set $P(W|X)$ leads to the mass constraint DA clustering algorithm; and the maximization of (2) against the source pmf set $P(X)$ is critical for the robust estimation to reject the outliers or noise.

2.1 Minimization of Mutual Information

For any fixed unconditional *a priori* pmf $p^*(x_j) \in P^*(X)$ (normally as an equal distribution in DA approach [8]), the objective function of minimization optimization is given by

$$\begin{aligned} F_{Min} = \min_{P(W|X)} \{ & \sum_{j=1}^{l}\sum_{k=1}^{c} p^*(x_j)p(w_k|x_j)d(x_j,w_k) \\ & + T\sum_{j=1}^{l}\sum_{k=1}^{c} p^*(x_j)p(w_k|x_j)\log\frac{p(w_k|x_j)}{p(w_k)} \} \end{aligned} \tag{6}$$

It turns out [8] that the resultant distribution is the titled distribution and is given by

$$p(w_k|x_j) = \frac{p(w_k)e^{-d(x_j,w_k)/T}}{\sum\limits_{k=1}^{c} p(w_k)e^{-d(x_j,w_k)/T}} \tag{7}$$

with

$$p(w_k) = \sum_{j=1}^{l} p^*(x_j)p(w_k|x_j) \tag{8}$$

The Lagrange multiplier T is referred as the temperature parameter to control data clustering procedure as its value is lowered from infinity to zero [8].

Plugging (7) back into (6), the optimal objective function becomes the well-know energy function as follows

$$F^*_{Min} = -T \sum_{j=1}^{l} p^*(x_j) \log \sum_{k=1}^{c} p(w_k) e^{-\frac{d(x_j, w_k)}{T}} \tag{9}$$

Minimize the above equation with respect to the prototype w_k, we have (detailed derivation is omitted)

$$p_k = \begin{pmatrix} -2v_k \\ v_k^T v_k - r_k^2 \end{pmatrix} = -\frac{1}{2} \frac{\sum\limits_{j=1}^{l} p^*(x_j)p(w_k|x_j)q_j}{\sum\limits_{j=1}^{l} p^*(x_j)p(w_k|x_j)M_j} \tag{10}$$

Note the cluster center v_k and radius r_k are simultaneously updated by the above equation.

DA optimization begins by determining the minimum of the free energy F^*_{Min} at high values of T and attempts to track the minimum through lower values of T, until the global minimum of the free energy at $T \to 0$ coincides with the global minimum of the original cost function $D(X, W)$. It is known for DA approach that during the annealing splits in the cluster representation occur [8]. These splits are related to qualitative changes in the optimization problem and have to be taken into account in the annealing process. The critical temperature for phase transition of *kth* cluster can be determined by the maximum eigenvalue of the spherical shell based covariance matrix

$$T^*_k = 2\lambda_{max}(C_k(X, W)) \tag{11}$$

where $C_k(X, W)$ is the covariance matrix of *kth* cluster

$$C_k(X, W) = \sum_{j=1}^{l} p^*(x_j|w_k)(x_j - v_k - r_k \frac{x_j - v_k}{\|x_j - c_k\|})^2 \tag{12}$$

where $p^*(x_j|w_k)$ is given by Bayes formula, see (17).

2.2 Maximization of Mutual Information

From the constrained minimization of MI in last subsection, we have obtained an optimal conditional probability, i.e., likelihood $\bar{p}(w_k|x_j)$. According to information theory, the mutual information can also be maximized against a prior

$p(x_i)$ to gain the maximum capacity in a noisy channel [9] [1], i.e., to maximize the following objective function

$$F_{Max} = \max_{P(X)}\{\sum_{j=1}^{l}\sum_{k=1}^{c} p(x_j)\bar{p}(w_k|x_j)\log\frac{p(x_j|w_k)}{p(x_j)} - \lambda\sum_{j=1}^{l} p(x_j)e_j\} \tag{13}$$

where $\lambda \in [0, +\infty)$ is a separated parameter to control the degree of robustness against outliers, e_j is the expense of using the jth input point, which is given by

$$e_j = \sum_{k=1}^{c} \bar{p}(w_k|x_j)d(x_j, w_k) \tag{14}$$

From [1] [2], we know the resultant prior that minimizes F_{Max} is given by

$$p(x_j) = \frac{p(x_j)c_j}{\sum\limits_{j=1}^{l} p(x_j)c_j} \tag{15}$$

where c_j is the capacity of the jth input point, which is defined by

$$c_j = exp\left(\sum_{k=1}^{c}\bar{p}(w_k)|x_j)\log\frac{\bar{p}(w_k|x_j)}{\bar{p}(w_k)} - \lambda e_j\right) \tag{16}$$

with the posteriori $p(x_j|w_k)$ obtained through the Bayes formula

$$p(x_j|w_k) = \frac{p(x_j)\bar{p}(w_k|x_j)}{\sum_{j=1}^{l} p(x_j)\bar{p}(w_k|x_j)} \tag{17}$$

Note that the mutual information (5) is not negative. However, the individual item in the sum of the capacity can be negative [1]. If the jth point x_j is taken into account and $p(w_k|x_j) < \sum\limits_{i=1}^{l} p(x_i)p(w_k|x_i)$, then the probability of the kth prototype is decreased by the observed point and gives a negative information about x_j. Then the particular input point may be considered as an unreliable point (outlier) and its negative effect must be offset by other input points. Therefore, the maximization of the mutual information (13) provides a good robust estimation of the noisy point (outlier) in term that the average information is over all clusters and input points. The robust estimation and optimization is to maximize the mutual information against the pmf $p(x_j)$ and $p(x_j|w_k)$, for any value of j, if $p(x_j|w_k) = 0$, then $p(x_j)$ should be set equal to zero in order to obtain the maximum, such that a corresponding point x_j can be deleted and dropped from further consideration in the optimization procedure as an outlier.

The MMI algorithm determines the degree of robustness and rejects the outlier (noisy input points) with maximum capacity by selecting a proper value of the parameter λ to pruning each input point. As discussed in [9], the robustness of the proposed algorithm will be increased if a bigger value of $\lambda > 0$ is selected, which implies that more outliers (noisy points) are detected in the robust estimation procedure under the constraint of each respective cluster. Note that the constraint in (13) is against the outliers surrounding the respective cluster and is not purposely designed for the mutual information maximization: if we set $\lambda = 0$, the MMI algorithm is only interesting in the rejection of inter-cluster outliers; if we let $\lambda > 0$, the algorithm is robust against both inter-cluster and intra-cluster outliers. The higher value λ is, the more data become outliers.

2.3 Optimal Cluster Number Selection

A natural question of clustering is how many clusters are appropriate for the description of a given data. Several cluster validity criteria specific for shell-type clusters, as e.g., shell thickness and shell hyper-volume, have been presented in the literature [5] [6] [4]. However, as discussed in [4], they may fail in some instances, especially when there exist noisy points in the data. This makes them impractical in real applications. We here refer to the statement in [9] to present a practical cluster validity criteria, based on the structural risk minimization (SRM) principle [10], to determine an optimal cluster number of spherical shells for a given data.

According to the information theory [2], a well designed communication channel should has few unreliable input data points to achieve the capacity. This implies that a good clustering algorithm with the correct cluster number should produce few outliers sitting between the underlining nature clusters. However, few outliers can only guarantee small empirical risk but not real risk (generalization) from the view of statistical learning theory. Increasing cluster number (as the temperature is lowered in MMI clustering) normally reduces the empirical error but increases the model complexity, a good cluster validity criteria should make a tradeoff between the empirical risk and model complexity, as the structural risk minimization (SRM) principle [10] does. We determine the optimal cluster number by the VC-bound [10] as follows,

$$V_b \leq \eta + \frac{\varepsilon}{2}(1 + (1 + \eta\frac{4}{\varepsilon}))^{1/2} \tag{18}$$

with

$$\eta = l_o/l \tag{19}$$

$$\varepsilon = \frac{h_c(\log\frac{2l}{h_c} + 1) - \log\frac{\zeta}{4}}{l} \tag{20}$$

where l_o is the number of outliers identified in the maximization optimization in the last subsection. $\zeta < 1$ is a constant. The VC-dimension of the complexity control parameter h_c is equal to the number of parameters, i.e., $h_c = c \times (n+1)$. The signal to noise ratio η in (19), appeared as the first term of the right hand side of the VC bound (18), represents the empirical risk and the second term is the confidence interval of the SRM based estimation. The novel cluster validity criteria is stated as: by evaluating the estimated VC-bound for each choosing cluster number by equation (18), we select the one that yields the minimum value of V_b as the optimal cluster number.

2.4 Pseudo-Code of MMI Algorithm

Based on the implementations of DA approach in [8] and capacity maximization in [1], we give the detailed pseudo-code of MMI for clustering of spherical shell-shaped data as follows.

- Step 1) Set the maximum number of clusters c_{max}, initial temperature $T_{ini} > 2T_1^*$ (see (11)), minimum temperature $T_{min} = T_{ini}/1000$, convergence parameter $\varepsilon = 0.001$, mass probability $p(w_1) = 1$, source distribution $p(x_j) = \frac{1}{l}$ $(j = 1, 2, \ldots, l)$, and $c = 1$.
- Step 2) Alternatively update (7) (8) and (10) for k = 1,2,...,c (fixed point iteration) until the maximum change in the prototypes between consecutive iterations is less than the given threshold value ε.
- Step 3) Set the control parameter $0 \leq \lambda < \infty$, and initialize distribution $p(x_j) = \frac{1}{l}$ $(j = 1, 2, \ldots, l)$, iteratively update (15) and (16) until

$$\log \max_{j=1,\ldots,l} c_j - \log \sum_{j=1}^{l} p(x_j) c_j < \varepsilon$$

  is satisfied.
- Step 4) Verify the robust solutions of the MMI algorithm around the optimal saddle point for a minimum value of the VC-bound (18) within the range of maximum cluster number c_{max}. If the minimum is found, then delete outliers, set $T \to 0$ for the titled distribution to obtain the probability of all data points for a hard clustering solution. Recalculate the cluster prototypes without outliers, then stop. Otherwise, go to next step.
- Step 5) If $T < T_{min}$ then stop. Otherwise, let $T = \eta T (0 < \eta < 1)$, and check condition for phase transition for $k = 1, 2, \ldots, c$, if critical T_k^* is reached for cluster k (see (11)), add a new cluster prototype by $p_{k+1} = p_k + \delta$ with $p(w_{k+1}) = p(w_k)/2$ and $p(w_k) = p(w_k)/2$, where δ is a small disturbance, let $c \leftarrow c + 1$, then go to step 2.

3 Experimental Results

We show the effectiveness of MMI by several simulation results. In all simulation results, the original data sets are marked by “o” and the partitioned clusters are displayed by using different marks. The circumferences (determined by centers and radii) are also plotted by dotted lines “...” for partition result visualization.

3.1 Effectiveness of Clustering Spherical Shells

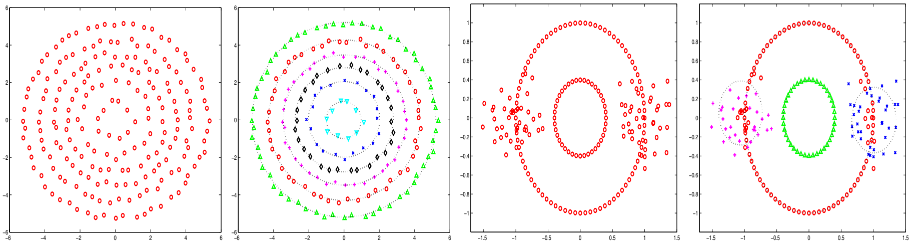
We show the effectiveness of MMI for clustering spherical shells by investigating several complicated cases, which are difficult for existing clustering techniques like FCS and FCSS algorithm to deal with. Two data sets are considered: the first one contains six disturbed ring data with identical center but different radii as shown in Fig.1a; the other one is the combination of compact spherical and ring-shaped clusters as shown in Fig.1c. The optimal cluster number determined by VC-bound is 6 for the first data and 3 for the second data, see next example for detailed discussion of cluster number selection. With the optimal cluster number, MMI correctly partitions each given data into the original data structure as shown in Fig.1b and Fig.1d respectively. From the view of classification, there are 0 errors in total 210 data points for the first data and 26 in total 200 data points for the second data. Note we don't eliminate the outliers in this example for visualization convenience. For the second data, the errors can be reduced using a scaled distance measure for compact spherical data points as discussed in [6].

3.2 Effectiveness of Determining Cluster Number

As discussed above, the optimal cluster number of spherical shells is auto-determined by VC-bound in MMI algorithm. In most cases, the value of λ is insensitive to the determination of the optimal cluster number. We set λ as a constant 0.5 in all experiments if it is not specified.

Concentric Shells. The concentric shell-shaped data consists of three disturbed spherical shells as shown in Fig.3a. We calculate the values of VC-bound V_b by equation (18) with different cluster number (from 2 to 6) for this data set as concluded in Fig.2. The partition results of MMI with $c = 2, 3, 4$ are also plotted in Fig.3b-d for visualization. As observed from Fig.2 and Fig.3, the VC-bound finds the optimal cluster number for this data set: V_b reaches minimum at $c = 3$, which corresponds to the best partition result.

Intersected Shells. The intersected shell-shaped data contains three disturbed spherical shells as shown in Fig.4a. We calculate the values of VC-bound V_b by



(a)Original Data 1 (b)MMI with $c = 6$ (c)Original Data 2 (d)MMI with $c = 4$

Fig. 1. Partition results of MMI with optimal cluster number for two specific data sets in the first example

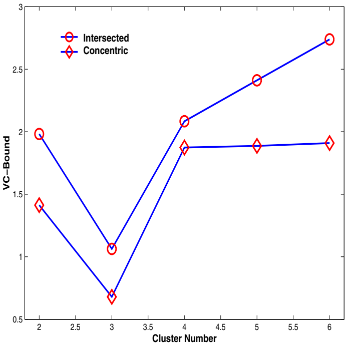
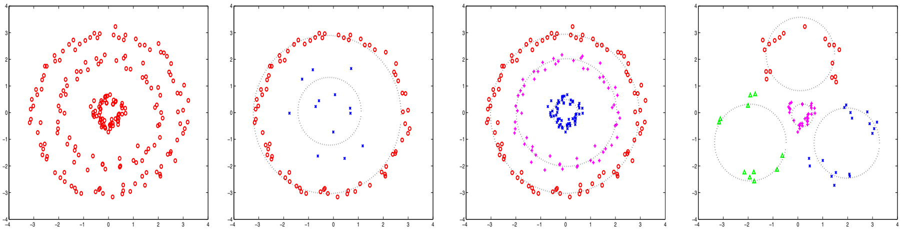
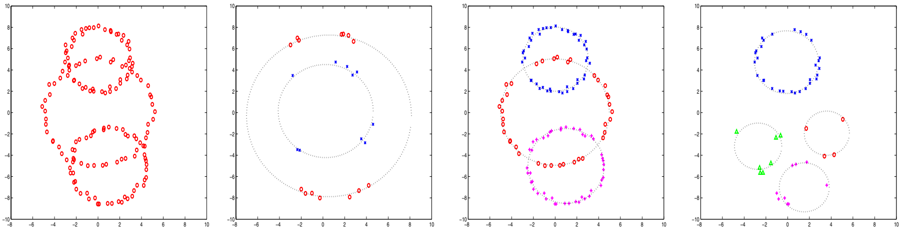


Fig. 2. VC-bound vs cluster number for the data sets in the second example. V_b reaches the minimum at $c = 3$ for both data sets.



(a)Original Data (b)$c = 2$, $V_b = 1.4128$ (c)$c = 3$, $V_b = 0.6796$ (d)$c = 4$, $V_b = 1.8739$

Fig. 3. Partition results and VC-bound of MMI with different cluster number c for concentric shell-shaped clusters in the second example. V_b reaches the minimum at $c = 3$ corresponding to the best result.



(a)Original Data (b)$c = 2$, $V_b = 1.9818$ (c)$c = 3$, $V_b = 1.0628$ (d)$c = 4$, $V_b = 2.0834$

Fig. 4. Partition results and VC-bound of MMI with different cluster number c for intersected shell-shaped clusters in the second example. V_b reaches the minimum at $c = 3$ corresponding to the most reasonable result.

equation (18) with different cluster number (from 2 to 6) for this data set as plotted in Fig.2. The partition results of MMI with $c = 2, 3, 4$ are also shown in Fig.4b-d for visualization. Similarly, the VC-bound reveals the optimal cluster number for this data set: V_b reaches minimum at $c = 3$, which corresponds to the most reasonable result.

4 Conclusion

A novel clustering algorithm for spherical shells detection and separation has been developed based on mini-max optimization of mutual information (MI). The new approach offers several improved features over existing clustering algorithms: First, it is independent of the data initialization and has the ability to avoid poor local optima due to the deterministic annealing (DA) process. Second, it provides a robust estimation of probability soft margin to phase out outliers though the maximization optimization of MI against input probability mass function (pmf). Finally, the optimal cluster number is estimated based on the structural risk minimization (SRM) principle of statistical learning theory. The superiority of the proposed clustering method has been tested by experimental results.

References

1. Blahut, R.E.: Computation of Channel Capacity and Rate-Distortion Functions, IEEE Tran. on Information Theory, Vol.18, (1972) 460-473
2. Blahut, R.E.: Princinple and practice of information theory, Addison-Wesley, (1987)
3. Dave, R.N.: Fuzzy shell-clustering and applications to circle detection in digital images, Int.J.General Systems, Vol.16, (1990) 343-355
4. Dave, R.N.: Validating fuzzy partitions obtained through c-shells clustering, Pattern Recognition Letters, Vol.17, (1996) 613-623
5. Krishnapuram, R., Nasraoui, O. and Frigui, H.: The fuzzy c-spherical shells algorithm: a new approach, IEEE Trans. Neural Networks, Vol.3, (1992) 663-671
6. Man, Y. and Gath, I.: Detection and Separation of Ring-Shaped Clusters Using Fuzzy Clustering, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.16, (1994) 855-861
7. Rose, K., Gurewitz, E. and Fox, G.C.: Statistical mechanics and phase transitions in clustering, Physical Review letters, Vol.65, (1990) 945-948
8. Rose, K.: Deterministic annealing for clustering, compression, classification, regression, and related optimization problems, Proc. of IEEE, Vol.86, (1998) 2210-2239
9. Song, Q.: A robust information clustering algorithm, Neural Computation, Vol.17, (2005) 2672-2698
10. Vapnik, V.N.: Statistical Learning Theory, John Wiley and Sons, Inc., NY, (1998)

Clustering of Interval-Valued Symbolic Patterns Based on Mutual Similarity Value and the Concept of *k*-Mutual Nearest Neighborhood

D.S. Guru and H.S. Nagendraswamy

Department of Studies in Computer Science, University of Mysore,
Mansagangothri, Mysore -570006, India
guruds@lycos.com, swamy_hsn@yahoo.com

Abstract. In this paper, a novel similarity measure for estimating the degree of similarity between two symbolic patterns, the features of which are of interval type is proposed. A method for clustering data patterns based on the mutual similarity value (MSV) and the concept of *k*-mutual nearest neighbourhood is explored. The concept of mutual nearest neighbourhood exploits the mutual closeness possessed by the patterns for clustering thereby providing the naturalistic proximity characteristics of the patterns. Experiments on various datasets have been conducted in order to study the efficacy of the proposed methodology.

Keywords: Symbolic data analysis, Interval-valued data, *k*-mutual nearest neighbourhood.

1 Introduction

Clustering plays a significant role in several exploratory pattern analysis, grouping, decision-making, machine learning situations, including data mining, document retrieval, image segmentation and pattern classification. Since similarity is the fundamental notion in clustering of data, a measure of the similarity between two patterns drawn from the same feature space is essential to most of the clustering procedures [11]. Among the existing clustering methodologies, the similarity-based clustering is a simple but powerful one. The guiding principle of similarity-based clustering is "similar patterns are within the same cluster and dissimilar patterns are in different clusters".

Several clustering algorithms [11] have been proposed for clustering conventional data sets of type crisp. However, their application on realistic data may not always yield the desired output as, in reality, data sets can appear in the form of continuous ratio, discrete absolute, interval, modal, multivalued and also multivalued with weights [1], which are very much generic than the conventional data sets. Thus in order to make clustering models more realistic to handle complex data, the existing conventional distance measures have to be modified or new distance measures that can heed such complex data sets have to be proposed.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 234 – 243, 2006.

Many similarity and dissimilarity measures were proposed for both conventional as well as symbolic data sets. But a few measures for symbolic data of interval type have been proposed [2], [4], [5], [7], [8], [9], [12]. The methods proposed in [4], [5] take into consideration the span (indicates the relative sizes of feature values without referring to common parts between them), position (it indicates the relative positions of two feature values on real line) and content (a measure of common parts between two feature values). Later some of the drawbacks of [4], [5] were overcome in [7], [12]. The dissimilarity measure for interval-valued type data through hyper box model representation is proposed in [2]. Multivalued type proximity measure and the concept of mutual similarity dissimilarity values for clustering symbolic pattern has been explored in [8], [9].

It has been observed from the literature survey that most of the conventional agglomerative clustering techniques use similarity proximity matrix for clustering patterns. The agglomerative clustering merges two patterns, which possess high similarity value at each level and continues the merging process until either the desired number of clusters is obtained or all the patterns are put into one cluster. However, it has been shown that about 62% of the total numbers of individuals in random artificial and natural populations are in mutual pairs. This concept of "Mutual Nearest Neighbourhood" was introduced [3] and successfully used for agglomerative and disaggregate clustering. Thus it is very appropriate to cluster patterns by looking at their mutual nearness than just by looking at their proximity matrix. In the proposed methodology, the concept of mutual nearest neighbourhood has been extended such that two classes of patterns are clustered into one class if all the patterns of both the classes are *k*-mutually nearest neighbours.

In this paper, we have proposed a novel similarity measure for estimating the degree of similarity between two patterns described by interval type data. A method of clustering patterns based on the mutual similarity value (MSV) of patterns and the concept of *k*-mutual nearest neighbours is explored. The proposed clustering method is based on a two layer clustering strategy. During the first layer, a similarity proximity matrix for symbolic patterns based on the proposed similarity measure is obtained. A position matrix is created from the similarity proximity matrix based on the similarity rank of patterns. The *k*-mutually-nearest neighbours algorithm is applied on the position matrix to obtain clusters of patterns.

The paper is organised as follows. Section 2 presents a novel method for estimating the degree of similarity between two symbolic patterns and a method of clustering symbolic patterns based on the proposed similarity measure and the concept of k-mutually nearest neighbours. Section 3 presents the results of the experiments conducted on variety of data sets to reveal the efficiency of the proposed methodology. Section 4 presents a comparative study with few well-known methods and finally the conclusion is given in section 5.

2 Proposed Methodology

In this section we introduce a novel method of computing the degree of similarity among symbolic patterns whose features are of interval type. Subsequently, a novel method of clustering symbolic patterns based on the mutual similarity value (MSV) and the concept of *k*-mutually nearest neighbours is presented.

2.1 A Novel Similarity Measure

In this subsection, we explain in detail a new symbolic similarity measure which estimate the degree of similarity between two patterns described by interval valued features.

Let $F_i = \left\{ \lfloor f_{i1}^-, f_{i1}^+ \rfloor, \lfloor f_{i1}^-, f_{i2}^+ \rfloor, \ldots, \lfloor f_{in}^-, f_{in}^+ \rfloor \right\}$ and $F_j = \left\{ \lfloor f_{j1}^-, f_{j1}^+ \rfloor, \lfloor f_{j1}^-, f_{j2}^+ \rfloor, \ldots, \lfloor f_{jn}^-, f_{jn}^+ \rfloor \right\}$ be the two symbolic patterns described by n interval valued features. Here f^- is the lower limit and f^+ is the upper limit of the interval.

Since we use the concept of mutual similarity between patterns for clustering, we first estimate the degree of similarity ($S_{i \to j}$) from the pattern F_i to the pattern F_j and the degree of similarity ($S_{j \to i}$) from the pattern F_j to the pattern F_i. Then the mutual similarity value (MSV) between the patterns F_i and F_j is defined to be the average of $S_{i \to j}$ and $S_{i \to j}$.

The degree of similarity from F_i to F_j with respect to their lth feature component is estimated based on degree of overlapping between their lower and upper limits of the intervals. If the interval $\lfloor f_{il}^-, f_{il}^+ \rfloor$ describing the lth feature component of F_i is contained in the interval $\lfloor f_{jl}^-, f_{jl}^+ \rfloor$ describing the lth feature component of F_j then the degree of similarity from F_i to F_j is taken as 1, otherwise it is given by the average degree of similarity between their respective lower and upper limits.

Thus, the degree of similarity from F_i to F_j with respect to their l^{th} feature is given by

$$S_{i \to j}^{l} = \begin{cases} 1 & \text{if } \left(f_{il}^- \geq f_{jl}^-\right) \text{ and } \left(f_{il}^+ \leq f_{jl}^+\right) \\ \dfrac{1}{2}\left[\dfrac{1}{1+\left|f_{il}^- - f_{jl}^-\right| * \beta} + \dfrac{1}{1+\left|f_{il}^+ - f_{jl}^+\right| * \beta}\right] & \text{otherwise} \end{cases} \tag{1}$$

Similarly, degree of similarity from F_j to F_i with respect to their l^{th} feature is given by

$$S_{j \to i}^{l} = \begin{cases} 1 & \text{if } \left(f_{jl}^- \geq f_{il}^-\right) \text{ and } \left(f_{jl}^+ \leq f_{il}^+\right) \\ \dfrac{1}{2}\left[\dfrac{1}{1+\left|f_{il}^- - f_{jl}^-\right| * \beta} + \dfrac{1}{1+\left|f_{il}^+ - f_{jl}^+\right| * \beta}\right] & \text{otherwise} \end{cases} \tag{2}$$

where β is the normalising factor, which is set to 0.1 for absolute values of features >1 and set to 1.0 for absolute values of features <1.

The mutual similarity value (MSV) between F_i and F_j with respect to l^{th} feature is given by

$$MSV\left(F_{il}, F_{jl}\right) = \frac{\left\lfloor S^{l}_{i \to j} + S^{l}_{j \leftarrow i} \right\rfloor}{2} \tag{3}$$

It shall be noticed that if the intervals are one and the same then the MSV between them is 1 (maximum); otherwise the similarity value depends on the extent to which the intervals are separated. More the extent to which they are separated less shall be the degree of similarity.

Once the degree of mutual similarity between two patterns with respect their l^{th} feature is computed then the overall degree of similarity between the patterns F_i and F_j with respect to all *n* features is given by

$$Sim\left(F_i, F_j\right) = \frac{1}{n}\sum_{l=1}^{n} MSV\left(F_{il}, F_{jl}\right). \tag{4}$$

2.2 Clustering of Symbolic Patterns

In order to cluster symbolic patterns described by interval-valued features, we first compute the degree of mutual similarity among all symbolic patterns using the proposed similarity measure as explained in the section 2.1 and a similarity proximity matrix is obtained. A position matrix of size (*r* x *r*) for all *r* patterns is created based on the descending order of their similarity. The position matrix gives the nearness position of patterns in terms of their similarity rank. The concept of *k*-mutual nearest neighbours, based on their position in the position matrix, is employed to cluster patterns, which are *k*-mutually nearest neighbours. In a position matrix, if the pattern P_i is the *k*-nearest neighbour of the pattern P_j , and the pattern P_j is the *k*-nearest neighbour of the pattern P_i , then P_i and P_j are said to be *k*-mutually nearest neighbours. This idea of *k*-mutually-nearest neighbours is successfully applied in our clustering procedure. According to this idea, if *m* patterns are put into one cluster then all those *m* patterns must be *k*-mutually nearest neighbours. The value of *k* is set to 2 initially and incremented by 1 each time until either we get desired number of clusters or the all patterns are put into a single cluster. Thus the proposed method of clustering symbolic patterns can be algorithmically expressed as follows.

Algorithm: Clustering-Symbolic-Patterns
Input: Interval-valued symbolic patterns (P_1,P_2, P_3,…, P_r)
Output: Clusters of symbolic patterns (C_1, C_2,…,C_m)
Method:

1. Obtain similarity proximity matrix for *r* patterns using the proposed similarity measure.
2. Create a position matrix of size (*r* x *r*) for all patterns based on the descending order of their similarity.

3. Let $C=\{C_1, C_2, \ldots, C_r\}$, initially contain r number of clusters each containing individual pattern.
4. Set number of clusters (*noc*) to *r*.
5. Set *k* to 2.
6. Merge two clusters C_u and C_v, if all the patterns in C_u and C_v are *k*-mutual nearest neighbours.
7. Decrement *noc*, the number of clusters by 1.
8. Increment *k* by 1.
9. Repeat steps 6 to 8 until either the desired number of clusters is obtained or all the patterns are put into single cluster i.e., *noc*=1.

Algorithm Ends

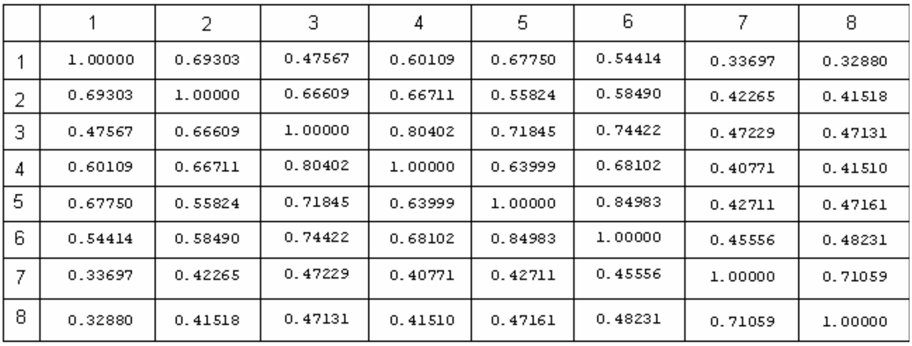
3 Experimental Results

We have conducted several experiments on variety of data sets to validate the efficiency of the proposed methodology. In this section we present the clustering results on a few well-known data sets.

3.1 Experiment 1

The first experiment is conducted on Ichino's fat oil data, which has been used by several researchers as a typical example of a data set involving interval-valued features. It is composed of eight patterns described by four interval-valued features [10]. The proposed similarity measure is employed on the data set to estimate the degree of similarity and the similarity matrix is obtained (see Table 1). Based on the similarity matrix, a position matrix, which gives the relative nearness of patterns in the descending order of similarity, is created (see Table 2). The concept of *k*-mutual nearest neighbourhood is employed on the position matrix and the dendrogram representation of the cluster formed is shown in the Fig. 1. A dendrogram is a special type of tree structure that provides a convenient picture of a hierarchical clustering [11]. It consists of layers of nodes, each representing a cluster and lines connecting nodes to represent clusters which are nested into one another. Cutting dendrogram horizontally creates a clustering.

Table 1. Similarity matrix for Fat Oil data



	1	2	3	4	5	6	7	8
1	1.00000	0.69303	0.47567	0.60109	0.67750	0.54414	0.33697	0.32880
2	0.69303	1.00000	0.66609	0.66711	0.55824	0.58490	0.42265	0.41518
3	0.47567	0.66609	1.00000	0.80402	0.71845	0.74422	0.47229	0.47131
4	0.60109	0.66711	0.80402	1.00000	0.63999	0.68102	0.40771	0.41510
5	0.67750	0.55824	0.71845	0.63999	1.00000	0.84983	0.42711	0.47161
6	0.54414	0.58490	0.74422	0.68102	0.84983	1.00000	0.45556	0.48231
7	0.33697	0.42265	0.47229	0.40771	0.42711	0.45556	1.00000	0.71059
8	0.32880	0.41518	0.47131	0.41510	0.47161	0.48231	0.71059	1.00000

Table 2. Pattern matrix obtained based on the similarity rank

Pattern Number	Similarity Positions							
	1	2	3	4	5	6	7	8
P_1	P_1	P_2	P_5	P_4	P_6	P_3	P_7	P_8
P_2	P_2	P_1	P_4	P_3	P_6	P_5	P_7	P_8
P_3	P_3	P_4	P_6	P_5	P_2	P_1	P_7	P_8
P_4	P_4	P_3	P_6	P_2	P_5	P_1	P_8	P_7
P_5	P_5	P_6	P_3	P_1	P_4	P_2	P_8	P_7
P_6	P_6	P_5	P_3	P_4	P_2	P_1	P_8	P_7
P_7	P_7	P_8	P_3	P_6	P_5	P_2	P_4	P_1
P_8	P_8	P_7	P_6	P_5	P_3	P_2	P_4	P_1

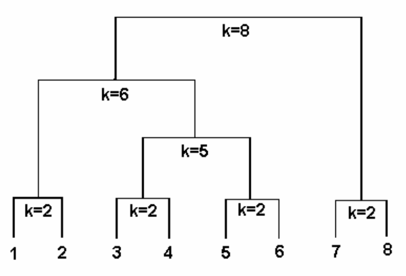


Fig. 1. Dendrogram representation of the clusters formation at various levels for Fat Oil data shown in the Table 1 by the proposed methodology

From Fig. 1, we can observe that the patterns {1, 2}, {3, 4}, {5, 6} and {7, 8} are 2-mutually nearest neighbors (k=2). That means, the pattern 2 is in the second position according to similarity rank for the pattern 1. Similarly, the pattern 1 is in the second position according to similarity rank for the pattern 2. One can notice this fact from the similarity matrix shown in Table 1and the position matrix shown in Table 2. Thus the patterns {1, 2} are clustered in the first level itself (dendrogram). The above argument is true for the patterns {3, 4}, {5, 6} and {7, 8}. The Patterns {3, 4} and {5, 6} are 5-mutually nearest neighbors (See Table 2) and hence they are clustered in the second level (for k=5) as shown in the dendrogram (Fig. 1). As there are no patterns, which are mutually nearest neighbors for k=3 and k=4, no patterns are merged for k=3 and k=4. The patterns {3, 4, 5, 6} and {1, 2} are 6-mutually nearest neighbors and are clustered at the level 3. No clusters are formed for (k=7) and finally we get a single cluster at the level 4 for k=8. We can cut the dendrogram at any level and realize the clusters depending on the desired number of clusters.

3.2 Experiment 2

We have also conducted an experiment on the data set given in [10] on microcomputers. The data set describes a group of microcomputers consisting of 12 patterns. Each pattern has five features. Two of the features are qualitative (Display and Microprocessor) and the rest are quantitative (RAM, ROM and Keys). The similarity proximity matrix for this data set is given in the Table 3. The dendrogram representation of the clusters formed at various levels is shown in Fig. 2. The proposed algorithm resulted in two clusters {1, 2, 3, 4, 5, 6, 8, 9, 10, 11,12} and {7}.

Table 3. Similarity matrix for Microcomputer data

	1	2	3	4	5	6	7	8	9	10	11	12
1	1.00000	0.91429	0.70804	0.70595	0.42657	0.48797	0.29192	0.52454	0.83926	0.96234	0.79567	0.58724
2	0.91429	1.00000	0.73916	0.79167	0.45768	0.51908	0.29979	0.61026	0.80718	0.93025	0.76359	0.61835
3	0.70804	0.73916	1.00000	0.55669	0.50429	0.54676	0.43697	0.45559	0.83807	0.71499	0.60941	0.62372
4	0.70595	0.79167	0.55669	1.00000	0.63959	0.62860	0.34804	0.71978	0.59884	0.72192	0.84136	0.70882
5	0.42657	0.45768	0.50429	0.63959	1.00000	0.91765	0.43089	0.69314	0.40420	0.43351	0.55682	0.76388
6	0.48797	0.51908	0.54676	0.62860	0.91765	1.00000	0.43281	0.77549	0.46561	0.49491	0.60444	0.80192
7	0.29192	0.29979	0.43697	0.34804	0.43089	0.43281	1.00000	0.28654	0.41715	0.29407	0.34471	0.46404
8	0.52454	0.61026	0.45559	0.71978	0.69314	0.77549	0.28654	1.00000	0.51120	0.54051	0.65003	0.57741
9	0.83926	0.80718	0.83807	0.59884	0.40420	0.46561	0.41715	0.51120	1.00000	0.87692	0.71026	0.47110
10	0.96234	0.93025	0.71499	0.72192	0.43351	0.49491	0.29407	0.54051	0.87692	1.00000	0.83333	0.59418
11	0.79567	0.76359	0.60941	0.84136	0.55682	0.60444	0.34471	0.65003	0.71026	0.83333	1.00000	0.75132
12	0.58724	0.61835	0.62372	0.70882	0.76388	0.80192	0.46404	0.57741	0.47110	0.59418	0.75132	1.00000

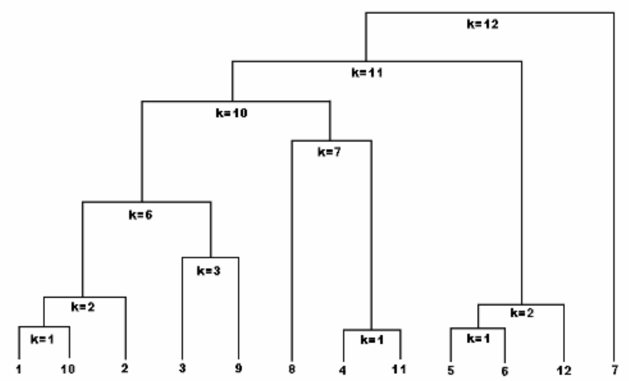


Fig. 2. Dendrogram representation of the clusters formation at various levels for Microcomputer data shown in the Table 4 by the proposed methodology

3.3 Experiment 3

In order to study the effectiveness of the proposed clustering methodology, we have conducted an experiment to cluster cities based on the temperature data used in [8], [9]. According to human observers [8], [9], two types of clusters are suggested.

Type1

Cluster-1: {2, 3, 4, 5, 6, 8, 11, 12, 15, 17, 19, 22, 23, 29, 31}
Cluster-2: {0, 1, 7, 9, 10, 13, 14, 16, 20, 21, 24, 25, 26, 27, 28, 30, 33, 34, 35, 36}
Cluster-3: {18} **Cluster-4**: {32}

Type2

Cluster-1: {2, 3, 4, 5, 6, 8, 12, 15, 17, 18, 19, 22, 23, 29, 31}
Cluster-2: {0, 1, 7, 9, 13 14, 16, 21, 24, 25, 26, 27, 28, 33, 34, 35, 36}
Cluster-3: {10} **Cluster-4**: {11} **Cluster-5**: {18} **Cluster-6**: {20}
Cluster-7: {30} **Cluster-8**: {32}

The proposed algorithm produced two clusters; cluster1: {2, 3, 4, 5, 6, 8, 11, 12, 15, 17, 18, 19, 22, 23, 29, 31} and cluster2: {0, 1, 7, 9, 10, 13, 14, 16, 20, 21, 24, 25, 26, 27, 28, 30, 33, 32, 34, 35, 36}. It can be observed that the method has produced clusters of Type1 as mentioned above. But the method could not classify cities 18 and 32 into separate clusters according to human observers' criteria. However, these cities are merged with cluster1 and cluster2 respectively at later stage. When we observe the temperature variation of Mauritius and Manila [8], [9], one can accept the possibility of classifying Mauritius (city 18) into a cluster where Manila (city 17) has already been classified. Similar argument is applicable to Tehran (city 32) as its temperature variation is similar to Frankfurt (city 9). Though our method could not produce finer classification as suggested by the human observers, the clusters obtained by the proposed methodology agree with the clusters suggested by the panel of human observers except cities 18 and 32. Hence, this experiment reveals the realistic nature of the proposed methodology.

4 Comparison and Discussion

In order to validate the correctness, we have compared the results of the proposed method with that of other available methodologies. For this purpose we have considered six methodologies which are listed in Table 4.

Table 4 summarizes the results obtained through the applications of all the seven methodologies including the proposed methodology on two different data sets viz., Fats and Oils data and Microcomputer data, as the results on these two data sets are studied by all the six methodologies. Since the temperature data is not considered by all the six methodologies, it is not included in Table 4. However, the experimentation has revealed that on temperature data, the proposed method has high consistency with human perception.

It can be noticed in Table 4 that the fats and oils patterns are either grouped into 2 clusters or into 3 clusters. The methods [5], [7], [8] have grouped the patterns into 2

Table 4. Results based comparison

Methodology	Fats and oils		Microcomputer	
	Description at 2 Clusters level	Description at 3 Clusters level	Description at 2 cluster level	Description at level more than or equal to 3 clusters
Ichino and Yaguchi (1994)	{1,2,3,4,5,6} {7,8}	{1,2} {3,4,5,6} {7,8}	{1,2,3,4,5,6,8,910,11,12} {7}	{1,2,3,9,10} {4,5,6,8,11,12} {7}
Gowda and Ravi (1995(a))	Not available	{1,2} {3,4,5,6} {7,8}	Not available	{1,2,4,6,8,9,10,11,12} {3} {7} {5}
Gowda and Diday (1991)	Not available	{1,2} { 3,4,5,6} {7,8}	Not available	{ 1,2,4,10,11} {7} {3,9} {5,6,12} {8}
Gowda and Diday (1992)	{1,2,3,4,5,6} {7,8}	Not available	Not available	{1,2,10,11} {7} {3,9} {4,5,6,8,12}
Gowda and Ravi (1995(b))	{1,2,3,4,5,6} {7,8}	Not available	{1,2,3,4,5,6,8,9,10,11,12} {7}	Not available
Guru et al. (2004)	{1,2,3,4,5,6} {7,8}	{1,2} {3,4,5,6} {7,8}	{1,2,3,4,5,6,8,9,10,11,12} {7}	{1,2,3,4,9,10,11} {4,5,7} {6} {11}
Proposed method	{1,2,3,4,5,6} {7,8}	{1,2} {3,4,5,6} {7,8}	{1,2,3,4,5,6,8,910,11,12} {7}	{1,2,3,4,8,9,10,11} {7} {5,6,12}

clusters ({1,2,3,4,5,6},{7,8}) and the methods [6], [4] have grouped the patterns into 3 clusters ({1,2}, {3,4,5,6}, {7,8}) based on their own cluster indicator function which acts as a stopping criterion. The entries not available in the Table 4 denote that the corresponding result has not been shown in the respective research work. We have not computed the same during experimentation as those methodologies require a prior knowledge of the number of samples in each pattern, which is indeed a real drawback of those approaches. Authors [10] have given the clustering of samples grouped into 2 clusters and as well as the samples grouped into 3 clusters. When our method is employed on the fats and oils data and the dendrogram (Fig. 2) is cut at the level of 3 clusters, the results are same as that of all the methods which yield 3 clusters [10], [6], [4], [8] and when agglomeration is allowed to continue upto 2 clusters then the result obtained is exactly same as that of the methods which yield 2 clusters [10], [5], [7], [8]. This shows consistency in the results of all the considered and our method on the fats and oils data.

It can also be noticed from Table 4 that the results obtained on Microcomputer data through all the 6 approaches are entirely different except the results of the methods [10], [8] and [7]. In the work [5], it is stated that no consistency can be expected on Microcomputer data. However, our method has resulted with 2 clusters, which are same as that of the methods [10], [8] and [7] encouraging their results.

5 Conclusion

In this paper, a new similarity measure useful for clustering of symbolic patterns is proposed. The concept of *k*-mutually nearest neighbours used for clustering patterns provides a realistic insight into the closeness among patterns in a cluster. One can accept the idea that two set of patterns are grouped together to form a single cluster only when all the patterns in both the clusters are *k*-mutually nearest neighbours and this fact can be revealed by the results of the proposed methodology on standard data sets. The efficacy of the proposed methodology is experimentally established and its validity is tested by comparing with the well-known methodologies.

References

[1] Bock H.H and E. Diday (Eds.), 2000. "Analysis of symbolic data". Springer verlag publication.

[2] Denoeux T. and M. Masson, 2000. "Multidimensional scaling of interval valued dissimilarity data". Pattern Recognition Letters 21, pp 83-92.

[3] Gowda K.C. and G. Krishna, 1977. "Agglomerative clustering using the concept of Mutual Nearest Neighborhood", Pattern Recognition, Vol. 10, pp. 105-112.

[4] Gowda K.C and E. Diday, 1991. "Symbolic Clustering using a new dissimilarity measure". Pattern Recognition, Vol 24, No 6, pp 567-578.

[5] Gowda K.C and Diday E., 1992. "Symbolic clustering using a new similarity measure". IEEE Trans SMC Vol 22, No 2, pp 368-378.

[6] Gowda K.C. and Ravi T.V., 1995 (a). "Agglomerative clustering of Symbolic Objects using the concepts of both similarity and dissimilarity". Pattern Recognition Letters 16, pp 647-652.

[7] Gowda K.C. and Ravi T.V., 1995 (b). "Divisive Clustering of Symbolic Objects using the concepts of both similarity and dissimilarity". Pattern Recognition, Vol 28, No 8, pp 1277-1282.

[8] Guru D.S., Bapu B. Kiranagi , P. Nagabhushan, 2004. "Multivalued type proximity measure and concept of mutual dissimilarity value for clustering symbolic patterns, Pattern Recognition. Vol. 38, No. 1, pp. 151-156.

[9] Guru D.S., Bapu B. Kiranagi, 2005. "Multivalued type dissimilarity measure and the concept of mutual similarity value useful for clustering symbolic patterns, Pattern Recognition Letters. Vol. 25, No. 10, pp. 1203-1213.

[10] Ichino M and H. Yaguchi, 1994. "Generalized Minkowski metrices for mixed feature type data analysis". IEEE Trans on System, Man and Cybernetics. Vol 24, No 4, April.

[11] Jain A.K. and C.R. Dubes, 1988. "Algorithms for Clustering Data". Prentice Hall, Engle Wood Cliffs.

[12] Prakash S.H.N., 1998. "Classification of remotely sensed data: some new approaches". Ph.D. Thesis. University of Mysore. Mysore, India.

Multiple Similarities Based Kernel Subspace Learning for Image Classification

Wang Yan, Qingshan Liu, Hanqing Lu, and Songde Ma

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
P.O. Box 2728, Beijing 100080, China
{wyan, qsliu, luhq, masd}@nlpr.ia.ac.cn

Abstract. In this paper, we propose a new method for image classification, in which matrix based kernel features are designed to capture the multiple similarities between images in different low-level visual cues. Based on the property that dot product kernel can be regarded as a similarity measure, we apply kernel functions to different low-level visual features respectively to measure the similarities between two images, and obtain a kernel feature matrix for each image. In order to deal with the problems of over fitting and numerical computation, a revised version of Two-Dimensional PCA algorithm is developed to learn intrinsic subspace of matrix features for classification. Extensive experiments on the Corel database show the advantage of the proposed method.

1 Introduction

With image data growing rapidly, how to efficiently manage and browse images is an urgent and challenging problem. Image classification is an important technique for image browsing and retrieval in a large database [1], [2].

As a typical pattern recognition problem, image classification has two key issues, i.e., feature extraction and classifier selection based on extracted features. Most previous studies concentrate on designing the classifier, and directly take low-level visual cues as features, such as color, shape and texture [1], [2], [3]. Different feature vectors are concatenated end to end and form the feature vector of the image. The similarity between two images is usually measured in the Euclidean space with these feature vectors. Zheng, et al, propose using local preserving projection (LPP) [4], [5] to capture the local manifold of the images, but LPP is a linear approach in nature. Practically, as for some complex image data, especially nature images, there exist complex nonlinear variations, which will degrade the performance of the classification methods.

Kernel Principal Component Analysis (KPCA) is a good nonlinear analysis method, which is actually a nonlinear version of Principal Component Analysis (PCA). Its idea is to first map the input data into an implicit feature space by a nonlinear mapping, and then the data are analyzed in the implicit feature space [6]. KPCA has been widely used in practical data analysis [6], [7], [8]. The new method proposed in this paper is inspired by KPCA.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 244–253, 2006.

We first give a new perception of KPCA. It can be regarded as having two independent steps: kernel feature extraction and PCA on the kernel features [9]. Kernel feature vector of an image can be calculated as follows: construct an vector with kernel dot products between the image and all the training images first, and then center the vector by subtracting its mean value. Based on this perception, a scheme of matrix based kernel features for image classification is proposed. Since in image classification and retrieval, images are often described by multiple visual cues, such as color, shape and texture, if kernel dot products between two images on different visual cues are computed respectively, we can get a dot product vector between two images, and the kernel feature vector of an image becomes a kernel feature matrix. From the view that the kernel dot product being a similarity measure [10], [11], the kernel feature matrix provides a strategy to measure the multiple similarities between the image and training images, which should be more precise for image classification. In order to deal with the problem of the evaluation of eigenvectors, a revised version of Two-Dimensional PCA (2DPCA) [12] is developed to learn the intrinsic subspace of image feature matrices. Extensive experiments on the Corel database show that the proposed method has an encouraging performance.

The rest paper is organized as follows: a new perception of KPCA is given in Section 2, and we present the proposed image classification scheme in Section 3. The experiments are reported in Section 4, followed by the conclusions in Section 5.

2 Kernel Principal Component Analysis

The idea of KPCA is first to map the input data $\{\mathbf{x}_i\}_{i=1}^N$ into an implicit feature space F by a nonlinear mapping ϕ, and the PCA is performed in F to get the nonlinear principal components of the input data [6]. It is unnecessary to know the mapping ϕ explicitly, and we only need to calculate the dot product between implicit features vectors $\{\phi(\mathbf{x}_i)\}_{i=1}^N$ with a kernel function that satisfies Mercer's theorem [6]. Gaussian kernel is used in this paper for its popularity in image classification and retrieval [13], [14], and its definition is as follows:

$$k(\mathbf{x}_1, \mathbf{x}_2) = (\phi(\mathbf{x}_1) \cdot \phi(\mathbf{x}_2)) = \exp\left(-\gamma \left\|\mathbf{x}_1 - \mathbf{x}_2\right\|^2\right) . \tag{1}$$

For the following analysis, we define some symbols first. $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ is the training set. The matrix $\mathbf{\Phi}(\mathbf{X}) = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_N)]$ is the mapping of training set in implicit feature space F. The Gram matrix $\mathbf{K} = [\mathbf{K}_1, \mathbf{K}_2, \cdots, \mathbf{K}_N]$, where the column vector $\mathbf{K}_i$ is composed of dot products between $\phi(\mathbf{x}_i)$ and all the training set in F, i.e., $\mathbf{K}_i = (k(\mathbf{x}_i, \mathbf{x}_1), k(\mathbf{x}_i, \mathbf{x}_2), \cdots, k(\mathbf{x}_i, \mathbf{x}_N))$. It can be seen that $\mathbf{K}$ is symmetrical.

KPCA is equivalent to solving the problem of eigenvectors and eigenvalues of covariance matrix $\mathbf{C}$ of $\{\phi(\mathbf{x}_i)\}_{i=1}^N$ [6].

$$\mathbf{C} = \frac{1}{N-1}(\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1}_N)(\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1}_N)^T , \tag{2}$$

where $\mathbf{1}_N$is an $N \times N$matrix with each entry equals $1/N$. Let $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_E]$ denotes the unitary eigenvector matrix of $\mathbf{C}$, where $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_E$ are unitary eigenvectors corresponding to positive eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_E$, we get

$$\frac{1}{N-1}\mathbf{W}^T(\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1}_N)(\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1}_N)^T\mathbf{W} = \mathbf{\Lambda} , \tag{3}$$

where $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \cdots, \lambda_E)$.

Since any eigenvector $\mathbf{w}_i$ with eigenvalue λ_i must lie in the span of $\{\phi(\mathbf{x}_i)\}_{i=1}^N$ [6], we have

$$\mathbf{W} = (\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1}_N)\mathrm{A} , \tag{4}$$

where $\mathrm{A} = [\alpha_1, \alpha_2, \cdots, \alpha_E]$ is a $N \times E$ matrix called eigenvector expansion coefficient matrix, and $\mathbf{w}_i = (\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1}_N)\alpha_i$.

Combining (3) and (4), and because $\mathbf{\Phi}^T(\mathbf{X})\mathbf{\Phi}(\mathbf{X}) = \mathbf{K}$, we get

$$\frac{1}{N-1}\mathrm{A}^T(\bar{\mathbf{K}} - \bar{\mathbf{K}}\mathbf{1}_N)(\bar{\mathbf{K}} - \bar{\mathbf{K}}\mathbf{1}_N)^T\mathrm{A} = \mathbf{\Lambda} , \tag{5}$$

where $\bar{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N\mathbf{K}$. In fact, each column of $\bar{\mathbf{K}}$ equals the corresponding column of $\bar{\mathbf{K}}$ subtracting its mean value, so we call $\bar{\mathbf{K}}$ centered Gram matrix. Thus, the solution of equation (5) is equivalent to solving the eigenvectors and eigenvalues of the covariance of $\bar{\mathbf{K}}$. Finally we normalize $\alpha_1, \alpha_2, \cdots, \alpha_E$ in order to make $(\mathbf{w}_i \cdot \mathbf{w}_i) = 1$. Note that $\bar{\mathbf{K}} - \bar{\mathbf{K}}\mathbf{1}_N$ is symmetrical and all of its eigenvalues are nonnegative, it can be proved that any eigenvector α_i of $(\bar{\mathbf{K}} - \bar{\mathbf{K}}\mathbf{1}_N)(\bar{\mathbf{K}} - \bar{\mathbf{K}}\mathbf{1}_N)^T/(N-1)$ with eigenvalue λ_i is eigenvector of $(\bar{\mathbf{K}} - \bar{\mathbf{K}}\mathbf{1}_N)$ with eigenvalue $\sqrt{(N-1)\lambda_i}$. So the normalization condition is as follows:

$$\begin{aligned} 1 = (\mathbf{w}_i \cdot \mathbf{w}_i) &= ((\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1}_N)\alpha_i \cdot (\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1}_N)\alpha_i) \\ &= \alpha_i^T(\bar{\mathbf{K}} - \bar{\mathbf{K}}\mathbf{1}_N)\alpha_i \\ &= \sqrt{(N-1)\lambda_i}(\alpha_i \cdot \alpha_i) = \sqrt{\lambda_i'}(\alpha_i \cdot \alpha_i) \end{aligned} , \tag{6}$$

where λ_i' is the eigenvalue of $(\bar{\mathbf{K}} - \bar{\mathbf{K}}\mathbf{1}_N)(\bar{\mathbf{K}} - \bar{\mathbf{K}}\mathbf{1}_N)^T$.

As for test samples $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_L]$, their projections in KPCA subspace are

$$\mathbf{Y} = \mathbf{W}^T(\mathbf{\Phi}(\mathbf{T}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1'}_N) = \mathbf{W}^T\mathbf{\Phi}(\mathbf{T}) - \mathbf{W}^T\mathbf{\Phi}(\mathbf{X})\mathbf{1'}_N , \tag{7}$$

where $\mathbf{1'}_N$ is the $N \times L$ matrix with each entry equals $1/N$. Since all columns of $\mathbf{W}^T\mathbf{\Phi}(\mathbf{X})\mathbf{1'}_N$ are identical, it is irrelevant for classification problem when using Euclidean distance. Combining (4) and (7) we get

$$\mathbf{Y} = \mathbf{W}^T\mathbf{\Phi}(\mathbf{T}) = \mathrm{A}^T(\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{1}_N)^T\mathbf{\Phi}(\mathbf{T}) . \tag{8}$$

Define matrix $\mathbf{K}^{test} = (k(\mathbf{x}_i, \mathbf{t}_j))_{N\times L}$ for test points, and then we get

$$\mathbf{Y} = \mathrm{A}^T(\mathbf{K}^{test} - \mathbf{1}_N\mathbf{K}^{test}) = \mathrm{A}^T\bar{\mathbf{K}}^{test} , \tag{9}$$

where $\bar{\mathbf{K}}^{test} = \mathbf{K}^{test} - \mathbf{1}_N\mathbf{K}^{test}$.

From the description above, we can see that KPCA is equivalent to solving the eigenvectors and eigenvalues of covariance of centered Gram matrix $\bar{\mathbf{K}}$. So KPCA can be regarded as having two independent steps: kernel feature extraction and PCA on the kernel features. Dot product vector of an image is composed of kernel dot products between the image and all the training images, and then the kernel feature vector of the image is the dot product vector by subtracting its mean value. PCA is then performed on the kernel feature vectors to get the eigenvector expansion coefficient matrix A. The only difference is we need to normalize the coefficients according to (6). The projections of test data in KPCA subspace are $\mathbf{Y} = \mathrm{A}^T\bar{\mathbf{K}}^{test}$.

3 Matrix Based Kernel Feature for Image Classification

Based on this new perception of KPCA, we extend the kernel feature vectors of images to kernel feature matrices to measure multiple similarities between two images.

3.1 Matrix Based Kernel Feature

In KPCA, kernel dot product is used to capture the similarity between two images. But this description is not sufficient when the feature vectors of the images contain several kinds of low-level features, because it only tells the general similarity rather than individual ones in each kind of low-level visual cues. Since in image classification and retrieval, images are often represented by multiple visual cues, such as color, texture and shape, we perform kernel dot products on different visual cues respectively, and get a dot product vector between two images. This vector describes similarities in different visual cues rather than one general similarity between images. So the kernel feature vector of the image in KPCA becomes a kernel feature matrix. Assuming that there are p visual cues to represent images, the kernel feature matrix $\mathbf{M}_i$ of image is defined as follows:

$$\mathbf{M}_i = \begin{bmatrix} k_1(\mathbf{x}_i^1, \mathbf{x}_1^1) & k_2(\mathbf{x}_i^2, \mathbf{x}_1^2) & \cdots & k_p(\mathbf{x}_i^p, \mathbf{x}_1^p) \\ k_1(\mathbf{x}_i^1, \mathbf{x}_2^1) & k_2(\mathbf{x}_i^2, \mathbf{x}_2^2) & \cdots & k_p(\mathbf{x}_i^p, \mathbf{x}_2^p) \\ \vdots & \vdots & \ddots & \vdots \\ k_1(\mathbf{x}_i^1, \mathbf{x}_N^1) & k_2(\mathbf{x}_i^2, \mathbf{x}_N^2) & \cdots & k_p(\mathbf{x}_i^p, \mathbf{x}_N^p) \end{bmatrix}, \tag{10}$$

where k_p is the dot product kernel function for the p-th visual cue, $\mathbf{x}_i^j$ is the j-th visual cue of the i-th image. We call $\mathbf{M}_i$ the matrix based kernel feature.

From the view that the dot product kernel is a similarity measure function, the matrix based kernel feature provides a multi-similarity representation, i.e., we can get p levels of similarities between the image $\mathbf{x}_i$ and all the training images, which should be more precise than the kernel feature vector in traditional KPCA.

Corresponding to the centering of vector $\mathbf{K}_i$ in KPCA, we center each column of $\mathbf{M}_i$ in the same way and get centered kernel feature matrix $\bar{\mathbf{M}}_i$.

3.2 Revised Two-Dimensional PCA

Following the traditional KPCA, we have to reshape $\bar{\mathbf{M}}_i$ into a vector with $N \times p$ elements first, and then perform PCA. However this reshaping lead to expensive computation due to dimension increasing by p times. For example, if there are 1000 training samples and 10 similarities between two images, then the number of dimension becomes 10000. Fortunately, the eigenvectors can be calculated efficiently using the SVD techniques [15], [16], and the process of generating the covariance matrix is actually avoided. But it is difficult to evaluate the covariance matrix accurately due to its large size and relatively small number of training samples. The eigenvectors cannot be obtained accurately, since they are determined by the covariance matrix. We revise 2DPCA algorithm proposed in [12] to deal with this problem.

As opposed to conventional PCA, 2DPCA is based on 2D matrices rather than 1D vectors. That is, the centered kernel feature matrix does not need to be previously transformed into a vector, a covariance matrix can be constructed using kernel feature matrices directly.

Let $\{\mathbf{B}_i\}_{i=1}^{N}$ denote the training data, 2DPCA is to project $\mathbf{B}_i$ by a transform matrix $\mathbf{X}$:

$$\mathbf{Z}_i = \mathbf{B}_i \mathbf{X} \; . \tag{11}$$

Since the total scatter of the projected samples can be characterized by the trace of covariance matrix $\mathbf{G}$ of them, the following criterion is adopted to maximize the discriminating power of the projection $\mathbf{X}$:

$$J(\mathbf{X}) = trace(\frac{1}{M-1}\sum_{i=1}^{N}(\mathbf{Z}_i-\mu)^T(\mathbf{Z}_i-\mu)) = \mathbf{X}^T\mathbf{G}\mathbf{X} \; , \tag{12}$$

where μ is the mean matrix of all $\mathbf{Z}_i$s. The optimal projection $\mathbf{X}_{opt} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_E]$ is a matrix with each column as a unitary vector that maximize $J(\mathbf{X})$, i.e., the eigenvectors of $\mathbf{G}$ corresponding to the first several largest eigenvalues [17].

We may take centered kernel feature matrices $\left\{\bar{\mathbf{M}}_i\right\}_{i=1}^{N}$ as training set $\{\mathbf{B}_i\}_{i=1}^{N}$ and perform 2DPCA directly, or transpose $\left\{\bar{\mathbf{M}}_i\right\}_{i=1}^{N}$ to get $\left\{\bar{\mathbf{M}}_i^T\right\}_{i=1}^{N}$ first, and perform 2DPCA. The latter scheme is adopted in this paper for its similarity to KPCA. Because each column of $\bar{\mathbf{M}}_i$ corresponds to centered kernel feature vector $\bar{\mathbf{K}}_i$ in KPCA, and the projections of the samples in KPCA subspace are the inner products between $\left\{\bar{\mathbf{K}}_i\right\}_{i=1}^{N}$ and eigenvectors of covariance matrix $\mathbf{C}$. It seems more reasonable to calculate inner products between column of $\bar{\mathbf{M}}_i$ and eigenvectors of $\mathbf{G}$. Finally, the eigenvectors in $\mathbf{X}_{opt}$ are normalized according to:

$$\sqrt{\lambda_i}(\mathbf{X}_i \cdot \mathbf{X}_i) = 1 \; , \tag{13}$$

where λ_i is eigenvalue corresponding to eigenvector $\mathbf{X}_i$.

For test points $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_L]$, kernel feature matrix $\mathbf{M}^{test}$ and its centered version $\bar{\mathbf{M}}^{test}$ can be calculated, the projections of test points are:

$$\mathbf{Z} = \left(\bar{\mathbf{M}}^{test}\right)^T \mathbf{X} \; . \tag{14}$$

4 Experiments

4.1 Experimental Data

We test the proposed method on the Corel image database. Our dataset contains 6000 images with 60 categories randomly selected from the Corel database. Each category with 100 manually labeled images are used as ground truth.

Four kinds of visual features are used in this paper to represent the images: color histogram, color moments, wavelet based texture and orientation histogram. Color histogram is taken in HSV space with quantization of 8×4=32 bins on H and S channels. The first three moments from each of the three color channels are used for color moment. a 24-dimensional PWT based wavelet texture features and an 8-dimensional orientation histogram are contained to construct an 73-dimensional feature vector for each image. Each feature component is normalized, s.t. variance of each equals 1.

4.2 Experimental Results

In our experiments, five subspace learning algorithms including the proposed method are compared. They are:

PCA: This means performing PCA directly on original 73-dimensional visual features.

KPCA: The Gaussian kernel is used, and we investigate the kernel parameter in [0.001, 1] and find $\gamma = 0.08$ gives the best performance.

The proposed method: We note it as MSPCA for simplicity. Since each image is represented by four kinds of visual cues, we perform four kernel functions to compute the kernel dot products between two images on each visual cue respectively. For each image, a matrix based kernel feature is obtained. The revised 2DPCA is applied to these features then. For simplicity, we adopt four Gaussian kernels with same parameters. By investigation of the kernel parameter in [0.001, 1], the performance of MSPCA is maximized when $\gamma = 0.5$.

LKPCA: As mentioned above in 3.2, after matrix based kernel features are calculated, we can reshape them into vectors first, and then perform PCA on them. The kernel functions used here are the same as in MSPCA.

LPP: The code of LPP is downloaded from http://people.cs.uchicago.edu/∼xiaofei/. The number of nearest neighbors N is set to 10 as in [4].

For the output of the above algorithms, the nearest neighbor classifier is used for classification.

We test the algorithms on several different subsets of the database. Each subset is a mixture of k categories, where k varies between 2 and 10. For each category number k, 200 subsets are randomly selected from the database.

Two groups of experiments are designed. The first one is designed to compare the performance of the five algorithms. For each subset, all the images are used as training data, and 75 images of each category are used as gallery, and the rest

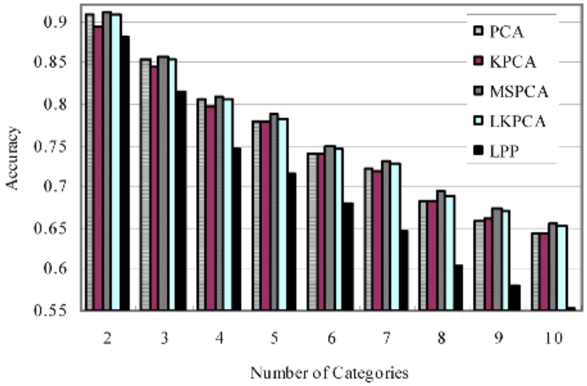


Fig. 1. Classification results comparison among PCA, KPCA, MSPCA, LKPCA and LPP

25% images are used as probe. Fig. 1 shows the experimental results, where the classification accuracies are their average on 200 subsets.

From Fig. 1, we can see that the propose algorithm, i.e., MSPCA has the best performance, followed by LKPCA, which uses the information of multiple similarities too. Since it is difficult to evaluate the covariance matrix accurately with relatively small training set in LKPCA, the accuracies are always lower than MSPCA. PCA outperforms KPCA when category number k varies from 2 to 8. Because the training set becomes more complicated with the increase of k, the performance of PCA is limited by its nature of linearity. That is why nonlinear KPCA has better results when k equals to 9 or 10. LPP fails in this experiment.

In order to further evaluate the performance of the proposed method, we conduct statistical tests between the proposed method and the other four methods. Since it is hard for us to know the distributions of the accuracies, non-parametric Wilcoxon's signed rank test (one-sided) for two related samples is adopted. We conducted tests between the results of the algorithms for each category number k respectively. The null hypothesis H_0 is the result of the proposed method has the same distribution as the result of algorithm A, where A is PCA, KPCA, LKPCA or LPP. The p-value of each tests are shown in Table 1.

Except for three tests (cells with italics), all p-values are less than 0.05, which means most of our tests show that there are significant differences between the accuracies of the proposed algorithm and four other algorithms respectively. Because the mean accuracies of MSPCA are always higher, MSPCA is considered better than other four algorithms at most of the time. Exceptions of the tests between PCA and MSPCA when k equals to 2 or 3 are probably due to the simplicity of training set, and adoption of kernel method such as MSPCA doesn't make much sense. But the average accuracies of MSPCA are still higher than those of PCA.

The second group of experiments is to test the generalization capability of the proposed method. 50 images of each category are used as training data and as gallery, and the rest 50% are used as probe. As in the first group of experiments,

Table 1. P-values of hypothesis tests between the proposed method and four other algorithms respectively based on the first type of experiments. "<2.2e-16" means "less than 2.2e-16".

k	PCA	KPCA	LKPCA	LPP
2	*0.1528*	1.060e-8	*0.09754*	6.965e-10
3	*0.05294*	5.566e-8	0.01381	<2.2e-16
4	0.005604	5.318e-9	0.02485	<2.2e-16
5	4.182e-7	5.039e-7	1.594e-7	<2.2e-16
6	2.737e-7	5.822e-8	0.004422	<2.2e-16
7	2.146e-8	4.92e-11	0.002100	<2.2e-16
8	1.777e-8	3.231e-10	0.0001069	<2.2e-16
9	<2.2e-16	3.272e-10	0.001384	<2.2e-16
10	4.113e-13	2.33e-11	0.001259	<2.2e-16

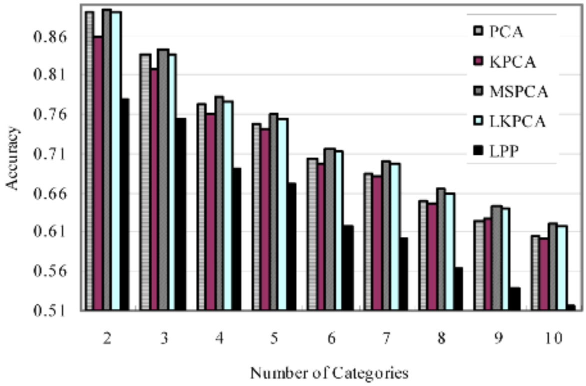


Fig. 2. Generalization capability comparison among PCA, KPCA, MSPCA, LKPCA and LPP

five algorithms are performed on 200 subsets and their average classification accuracies are calculated. The comparison of the result is shown in Fig. 2.

These experimental results are similar to the results of the first group. MSPCA is the best, followed by LKPCA. This shows the good generalization capability of MSPCA. When the number of categories is small (no more than 8), linear PCA outperforms KPCA. For more complicated data set, KPCA is preferred to PCA. LPP also fails.

We conducted Wilcoxon's signed rank tests between the results of the algorithms too, for each k respectively. The p-value of each tests are shown in Table 2.

All p-values but one (the cell with italics) are small enough to show that the performance of MSPCA is better than the other four methods when using out-of-sample data, which show its good generalization capability again. When number of categories k equals 2, test between PCA and MSPCA fails to reject the null hypothesis. It is probably due to the simplicity of the image set. But the mean accuracy of MSPCA is still higher than PCA.

Table 2. P-values of hypothesis tests between the proposed method and four other algorithms respectively based on the second type of experiments

k	PCA	KPCA	LKPCA	LPP
2	*0.07667*	<2.2e-16	0.001222	<2.2e-16
3	3.887e-5	<2.2e-16	3.031e-5	<2.2e-16
4	3.496e-7	<2.2e-16	2.951e-7	<2.2e-16
5	4.169e-12	<2.2e-16	8.771e-10	<2.2e-16
6	<2.2e-16	<2.2e-16	9.518e-8	<2.2e-16
7	<2.2e-16	<2.2e-16	3.009e-9	<2.2e-16
8	<2.2e-16	<2.2e-16	1.415e-12	<2.2e-16
9	<2.2e-16	<2.2e-16	6.566e-9	<2.2e-16
10	<2.2e-16	<2.2e-16	4.781e-10	<2.2e-16

5 Conclusions

In this paper, we conceive a new perception of KPCA, i.e., it can be regarded as having two separated steps: kernel features extraction and PCA based feature analysis. Dot product vector of an image is composed of kernel dot products between the image and all the training images, and then the kernel feature vector of the image is the dot product vector by subtracting its mean value. Based on this perception, we propose a new scheme of the matrix based kernel features for image clustering. With four kinds of visual cues, i.e., color histogram, color moment, wavelet based texture, and orientation histogram, we perform a dot product kernel to compute the similarity between two images respectively, and then obtain the matrix based kernel feature of an image with multi-similarities. In order to efficiently deal with the problem of the evaluation of eigenvectors, a matrix based KPCA algorithm is developed to learn the subspace of matrix features for classification. Extensive experiments on the Corel database are conducted to show the advantage of the proposed method.

Acknowledgement

This work is supported by the Natural Sciences Foundation of China Under Grant No 60135020, 60121302 and 60405005.

References

1. Chen, Y., Wang, J., Krovetz, R.: Content-based image retrieval by clustering. In: Proc. of ACM SIGMM Int. Workshop on Multimedia Information Retrieval. (2003)
2. Grodon, S., Greenspan, H., Goldberger, J.: Applying the information bottleneck principal to unsupervised clustering of discrete and continuous image representations. In: Proc. of Int. Conf. Computer Vision. (2003)
3. Barreno, M.: Spectral methods for image clustering. Tech-Report CS 218B, U.C. Berkeley (2004)

4. Zheng, X., Cai, D., He, X., Ma, W., Lin, X.: Locality preserving clustering for image database. In: Proc. of. ACM Multimedia. (2004)
5. He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing System. Volume 16., Cambridge, MA, MIT Press (2004)
6. Schölkopf, B., Smola, A., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation **10** (1998) 1299–1319
7. Yang, M., Ahuja, N., Kriegman, D.: Face recognition using kernel eigenfaces. In: Proc. of. Int. Conf. Image Processing. (2000)
8. Mika, S., Schölkopf, B., Smola, A., Müller, K., Scholz, M., Rätsch, G.: Kernel PCA and de-noising in feature spaces. In: Advances in Neural Information Processing System. Volume 11., Cambridge, MA, MIT Press (1999)
9. Liu, Q., Jin, H., Tang, X., Lu, H., Ma, S.: A new perception of kernel features. Tech-Report, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (2005)
10. Liu, Q., Liu, H., Ma, S.: Improving kernel fisher discriminant analysis for face recognition. IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Image and Video Based Biometrics **14** (2004) 42–49
11. Liu, Q., Huang, R., Liu, H., Ma, S.: Face recognition using kernel-based fisher discriminant analysis. In: Proc. of. Int. Conf. Automatic Face and Gesture Recognition. (2002)
12. Yang, J., Zhang, D., Frangi, A., Yang, J.: Two-Dimensional PCA: A new approach to appearance-based face representation and recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence **25** (2004) 131–137
13. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: Proc. of. ACM Multimedia. (2001)
14. Zhang, L., Lin, F., Zhang, B.: Support vector machine for image retrieval. In: Proc. of. Int. Conf. Image Processing. (2001)
15. Sirovich, L., Kirby, M.: Low-dimensional procedure for characterization of human faces. J. Optical Soc. Am. **4** (1987) 519–524
16. Kirby, M., Sirovich, L.: Application of the KL procedure for the characterization of human faces. IEEE Trans. on Pattern Analysis and Machine Intelligence **12** (1990) 103–108
17. Yang, J., Yang, J.: From image vector to matrix: A straightforward image projection technique—IMPCA vs. PCA. IEEE Trans. on Pattern Analysis and Machine Intelligence **35** (2002) 1997–1999

Boosted Algorithms for Visual Object Detection on Graphics Processing Units

Hicham Ghorayeb, Bruno Steux, and Claude Laurgeau

Ecole des Mines de Paris, Robotics Center, 60 bd Saint-Michel,
75272 Paris Cedex 06, France
{ghorayeb, steux, laurgeau}@ensmp.fr
http://caor.ensmp.fr

Abstract. Nowadays, the use of machine learning methods for visual object detection has become widespread. Those methods are robust. They require an important processing power and a high memory bandwidth which becomes a handicap for real-time applications. The recent evolution of commodity PC computer graphics boards (GPU) has the potential to accelerate those algorithms.

In this paper, we present a novel use of graphics hardware for object detection in advanced computer vision applications. We implement a system for object-detection based on AdaBoost [1]. This system can be tuned to run partially or totally on the GPU. This system is evaluated with two face-detection applications. Those applications are based on the boosted cascade of classifiers: Multiple Layers Face Detection (MLFD), and Single Layer Face Detection (SLFD). We show that the SLFD implementation on GPU performs up to nine times faster than its CPU counterpart. The MLFD, in the other hand, can be accelerated using the GPU and performs up to three times faster than the CPU.

To the best of our knowledge, this is the first attempt to implement a sliding window technique for visual object-detection on GPU, with promessing performance.

1 Introduction

Object detection is the ability to detect and localize objects within an image or a scene. One of the techniques used for object detection is called sliding-window. Sliding-window techniques allow a top-down approach. These techniques use a detection window of a fixed size and place this detection window over the input image at a given location. Then the algorithm determines whether the content of the image inside the window represents the object of interest or not. The search is repeated for all locations and scales of the input image, therefore the classification has to be very fast.

At the core of most sliding-window algorithms is usually a discriminative classifier, e.g., AdaBoost [1], Neural Network [2] or Support Vector Machine [3].

AdaBoost algorithm was first used by Viola & Jones in [4] for learning visual features for object detection. They demonstrated their method on a face detection application. Since, other families of features have been introduced, e.g.,

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 254–263, 2006.

illumination independent features [5], motion based features [6], Haar like features [7], etc. Those detectors are used to detect a wide range of objects, e.g., cars, pedestrians, bikes, etc. They are not always running in real time, depending on the complexity of the features used. The numerical complexity of the used features often imposes high demands on memory and computing resources. As a consequence, hardware acceleration of the detection core is starting to be a research area of interest [8].

The rapid increase in the performance of graphics hardware, coupled with recent improvements in its programmability, have made graphics hardware a compelling platform for computationally demanding tasks in a wide variety of application domains such as computer vision [9], scientific computation, and many more. An alternative mode of research is leading towards the implementation of general object detection core on computer graphics hardware.

The present work is demonstrated on, and in part motivated by, the task of face detection. The ability to detect faces in a scene is critical for humans in their everyday activities. Consequently, automating this task would be useful in many application areas such as intelligent human-computer interfaces, content based image retrieval, surveillance as well as many other areas.

The paper is organized as follows: design issues of a cascade of classifiers are presented in Section 2. Section 3 presents the GPU architecture and programming tools. Section 4 presents the implementation details on the GPU. Section 5 presents experimental results and section 6 concludes the paper.

2 Boosted Cascade of Classifiers

A cascade of classifiers (Fig. 1) is formed by combination of different numbers of simple weak classifiers with increasing complexity. The Cascade allows complexities of input patterns to be adapted. Using a simple classifier, face and non-face patterns are processed the same way. This leads to intensive consumption time for simple non-face patterns and makes the detection speed constant for every input, whatever its complexity. The main idea of building a cascade of classifiers is to overcome this weakness. Only input sub-windows that have passed through all layers of the cascade are classified as faces. With this structure, non-face patterns can be simply rejected by simple classifiers as shown in Table 1. Table 1 shows a comparison between a simple classifier and a cascaded classifier both achieving comparable detection rates. Note that cascaded algorithms can not perform as well as non cascaded algorithms, but their speed to success ratio is better [5].

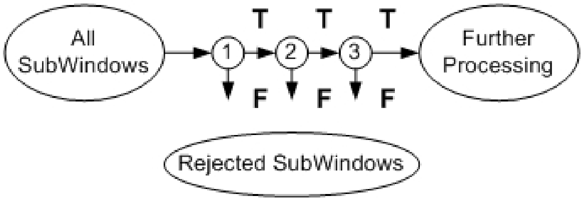


Fig. 1. Schematic of the detection cascade

Table 1. MLFD profiling: the detector has 12 layers. On a CPU , the total classification time for a 415x255 image is 175.12 ms. SLFD profiling: the detector has only one layer with 1600 features. On a CPU, the total classification time for a 415x255 image 18805.04 ms.

MLFD				SLFD			
Layer	Size	SubWindows	Time(ms)	Layer	Size	SubWindows	Time(ms)
0	2	92196	58.02	0	1600	92196	18805.04
1	3	28916	21.89				
2	5	16821	32.52				
3	10	6344	22.85				
4	21	2332	16.33				
5	21	804	7.93				
6	32	323	4.40				
7	48	143	2.54				
8	34	72	1.11				
9	58	49	0.62				
10	60	39	0.90				
11	1600	27	6.01				

2.1 The Control-Points Features

In this paper we implement boosted cascade of classifiers based on the Control-Points features proposed by Abramson [5]. They are working on gray images. Given an image of width W and height H (or a sub-window of a larger image, having these dimensions), we define a *control-point* to be an image location in the form $\langle i, j \rangle$, where $0 \leq i < H$ and $0 \leq j < W$. Given an image location z, We denote by $val(z)$ the pixel value in that location.

The Control-Points feature consists of two sets of control points, $x_1 \dots x_n$ and $y_1 \dots y_m$, where $n, m \leq K$. Each feature either works on the original $W \times H$ image, on a half-resolution $\frac{1}{2}W \times \frac{1}{2}H$ image, or on a quarter resolution $\frac{1}{4}W \times \frac{1}{4}H$

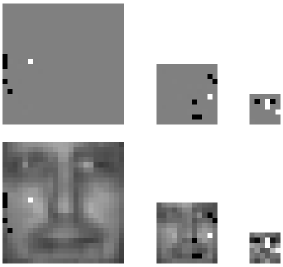


Fig. 2. The Control-Points features work on three resolutions (24x24, 12x12, 6x6) -in the case of face detection- and examine the image in single-pixel "control-points". The feature classifyies positively when all the pixel values in the black locations are higher than all the pixel values in the white locations.

image. These two additional scales have to be prepared in advance by downscaling the original image.

To classify a given image, a feature examines the pixel values in the control points $x_1 \dots x_n$ and $y_1 \dots y_m$ in the relevant image (original, half or quarter). The feature answers "yes" if and only if for every control point $x \in \{x_1 \dots x_n\}$ and every control point $y \in \{y_1 \dots y_m\}$, $val(x) > val(y)$. Some examples are given in Fig. 2.

3 Programmable Graphics Hardware

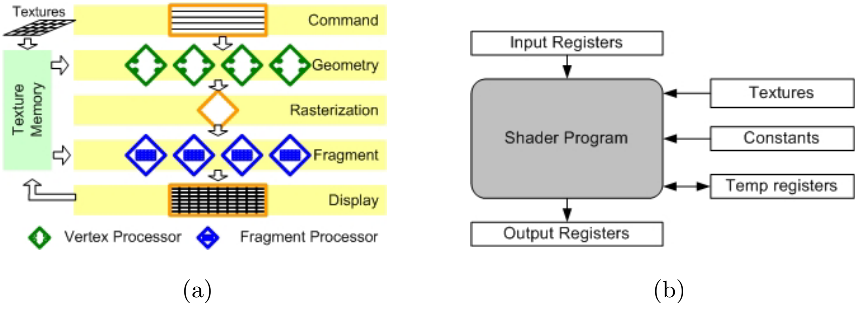
3.1 Graphics Pipeline

The GPU implements the different stages of the 3D graphics acceleration pipeline as shown in Fig. 3(a): command, geometry, rasterization, fragment and display.

Two stages are programmable: the geometry stage and the fragment stage. The geometry stage implements multiple vertex processors which perform geometric transformations and lighting operations on geometric primitives. Programs running on a vertex processor are called vertex shaders. After vertices are projected to screen space, the rasterizer calculates fragment information by interpolating vertex information. Then, the rasterizer assigns fragment rendering tasks to fragment processors. The fragment stage implements multiple fragment Processors. Programs running on the fragment processor are called pixel shaders. Fragment processor renders one fragment at a time. After a fragment has been rendered, the fragment processor writes the final color information into the fragment's designated location in the frame buffer for display.

3.2 Fragment Processor Architecture

The execution environment of a fragment (or vertex) processor is illustrated in Fig. 3(b). For every vertex or fragment to be processed, the shader program



(a) (b)

Fig. 3. (a)Reduced representation of the graphics pipeline. (b) Programming model for current programmable graphics hardware. A shader program operates on a single input element (vertex or fragment) stored in the input registers and writes the execution result into the output registers.

receives from the previous stage the graphics primitives in the read-only input registers. The shader is then executed and the result of rendering is written on the output registers. During execution, the shader can read a number of constant values set by the host processor, read from texture memory (latest GPUs started to add the support for vertex processors to access texture memory), and read and write a number of temporary registers.

3.3 Programming Language

There are two levels of programming languages that can be used to program graphics hardware: the assembly level shading language, and high level shading languages (such as the Cg language [10] from nVIDIA). These languages are not easy to use for non graphics programmer to implement general purpose computation. In Section 3.4 we present a platform that makes an abstraction of the graphics hardware as a slave co-processor, and a suitable model of computation.

3.4 Streaming Model of Computation

A streaming application is composed of two main objects: streams and kernels. A stream is a collection of records which require similar computation. A kernel is a program applied to each record of the input stream.

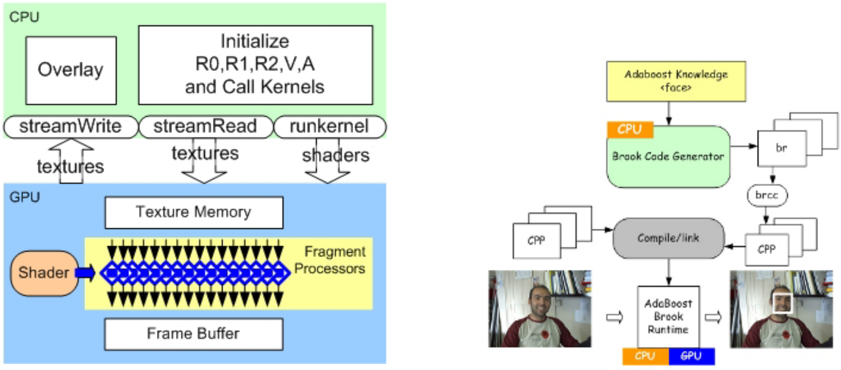
GPU can be considered as a stream co-processor [11][12]. Streams are mapped into textures and kernels are mapped into fragment shaders as shown in Fig. 4(a).

I. Buck, presents In [13] a programming environment for general purpose stream computing called Brook . Brook for GPU is the implementation of Brook for graphics hardware [13]. It consists in two components: a kernel compiler, which compiles kernel functions into legal Cg code, and a runtime system built on top of OpenGL [14] which implements the Brook API. Without Brook, stream management is performed by the programmer, requiring data to be manually packed into textures and transferred to and from the hardware. Kernel invocation requires the loading and binding of shader programs and the rendering of the geometry. This way of coding general purpose computation on GPU poses difficulties for non graphics programmers.

4 Implementation

We analyzed the code of a boosted cascade of classifiers based on Contol-Points features. Two main functional blocks related to the classification process can be highlighted:

- Internal preprocessing: the Control-Points features are applied to the three resolutions of the input fame. This operation is undertaken for each input frame.
- Binary classifier: this is a pixel wise operation. It consists in a sliding window operation. At each pixel location of the input frame, the possibility to have an object at this position defined by its upper-left corner and object dimensions is tested.



(a) (b)

Fig. 4. (a) Graphics pipeline used as streaming co-processor. Three services are available: StreamRead, StreamWrite and RunKernel. (b) Final System, Brook Runtime.

The internal preprocessing could be accelerated using a traditional API for OpenGLfor rendering a texture to a given Quad of size smaller than the dimensions of the given texture. This can be done using specific GL Options. The main part moved to the GPU is the binary classifier which consists in applying the same computation to each pixel in the input frame. Algorithm 1 presents the implementation of the cascade as a streaming application. The input streams are R_0, R_1, R_2, A and V. The different layers are called successively and the result of each layer is transmitted to the next layer using the V stream. This part of the implementation is running on the CPU (Fig. 4(a)). Algorithm 2 presents the pseudo-code of the implementation of the layer into several kernels (shaders) running on the GPU (Fig. 4(a)).

Algorithm 1. Cascade of Classifiers

```
Require: Intensity Image $R_0$
Ensure: a voting matrix
Build $R_1$ and $R_2$
Initialize $V$ from Mask
StreamRead $R_0$, $R_1$, $R_2$ and $V$
for all $i$ such that $0 \leq i \leq 11$ do
    $V \Leftarrow$ RunLayer $i, R_0, R_1, R_2, V$
end for
StreamWrite $V$
```

4.1 Hardware Constraints

The graphics hardware has many constraints on the shaders. On the graphics cards implementing NV30 shaders, shader size is limited to 1024 instructions and the constant registers are limited to 16. On more advanced graphics cards

Algorithm 2. Layer: Weighted Sum Classifier

Require: Iterator, R_0, R_1, R_2 and V
Ensure: a voting stream.

```
if $V$[Iterator] = true then
    $S \Leftarrow 0$
    for each feature $F_j$ in the layer do
        $A \Leftarrow$ RunFeature $j, R_0, R_1, R_2, V$
        $S \Leftarrow S + A*$ WeightOf($F_j$)
    end for
    if $S \leq$Threshold then
        return false
    else
        return true
    end if
else
    return false
end if
```

implementing NV40 shaders, shader size is unlimited, but the constant registers are limited to 32. These constraints affect the design of the application in terms of streaming application. Thus, the cascade of binary classifiers, has to be decomposed into several kernels, each corresponding to a layer. To achieve a homogeneous decomposition of the application, each layer is decomposed into several kernels, called scans, and the whole cascade is equivalent to a list of successive scans as shown in Fig. 5. Fig. 5 shows that the data transmission between layers is done using the V stream, and the intra-layer data transmission is done using the A stream to accumulate the intermediate features calculations.

4.2 Brook Implementation

Our code generator is implemented in the Perl script language to generate BrookGPU programs according to input AdaBoostlearning knowledge. The code of the application is generated in BrookGPU language, for two reasons:

- The code should be portable to various architectures, even future architectures that are not yet defined. Generating high-level language programs will allow fundemental changes in hardware and graphics API as long as the compiler and runtime for high-level language compilers keeps up with those changes.
- The transparency provided by Brookwhich makes it easier to write a streaming application. Fig. 4(b) shows the final system developped with BrookGPU.

The AdaBoost learning knowledge description serves as an input to the code generator which generates the Brook kernels as well as the main C++ code for streams initialization and read/run/write calls handling. The generated code has the ".br" extension, this file is compiled by brcc to generate the C++ code as well as the assembly language targeted to the GPU. This C++ file is then compiled and linked to the other libraries to build the executable.

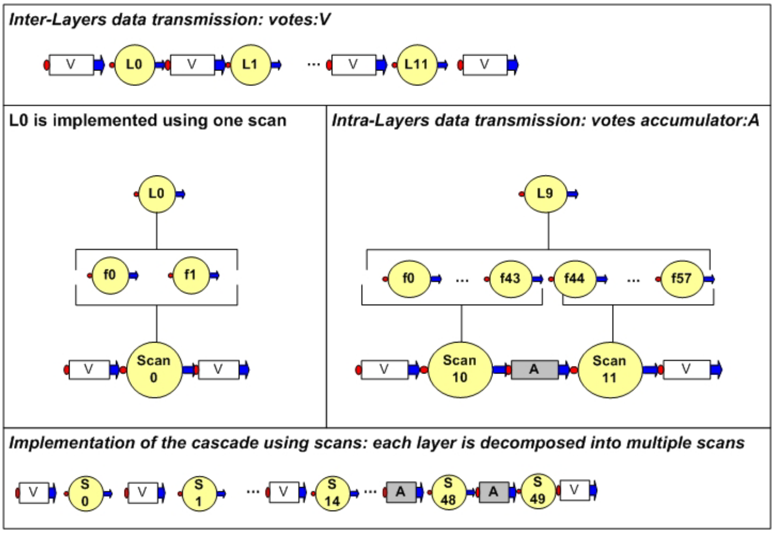

Inter-Layers data transmission: votes:V
V
L0
V
L1
...
V
L11
V
L0 is implemented using one scan
L0
f0
f1
V
Scan 0
V
Intra-Layers data transmission: votes accumulator:A
L9
f0
...
f43
f44
...
f57
V
Scan 10
A
Scan 11
V
Implementation of the cascade using scans: each layer is decomposed into multiple scans
V
S 0
V
S 1
...
V
S 14
...
A
S 48
A
S 49
V

Fig. 5. Multiscans technique applied to the MLFD detector. Layer L_0 is composed of two features; it is implemented using only one scan S_0. Layer L_9 is composed of 58 features; it is implemented using two scans S_{10} and S_{11}. S_{10} produces an intermediate voting result, and S_{11} produces the vote of L_9.

5 Performance Analysis

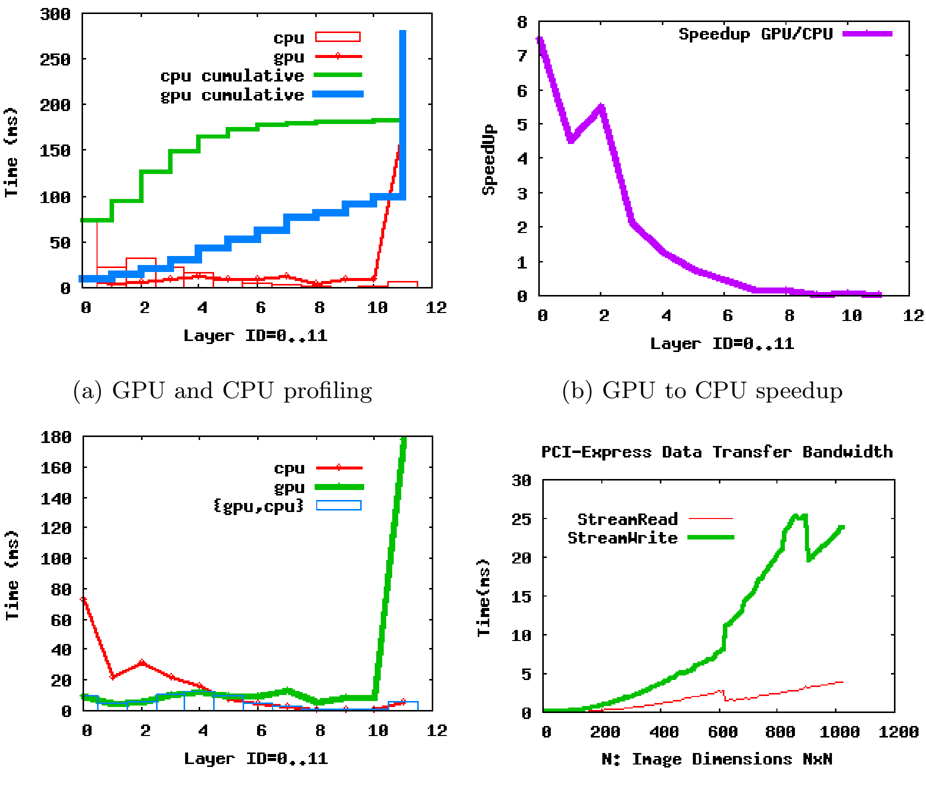
We tested the GPU implementation on a nVIDIA Geforce 6600GT on PCI-Express. The host is an athlon 64 3500+, 2.21 Ghz with 1G DDR. On this platform, the PCI-Express provides a hight data bandwidth. Three ms only are needed to read back a quarter pal image from the GPU (Fig. 6(d)).

5.1 Single Layer Face Detection

On a x86 processor, the SLFD requires 18.8s to classify 92k subwindows. The GPU implementation requires 2s to classify 100k subwindows. Thus, the GPU produces a speedup of 9.4 compared to the pure CPU implementation.

5.2 Multiple Layers Face Detection

The CPU version of the MLFD spends most of its time on the first 6 layers as shown in Fig. 6(a). This is because of the high number of windows to test at these layers is still considerable. In the last 6 layers, even if the layers are too complex, the number of windows to classify is not too high. Conversely, the GPU implementation spends most of its computation time on the last layers, and less time on the first 6 layers. This is because of the first layers, the number of features is not too high, so each layer requires a single pass on the GPU, which is very fast compared to the CPU. But, the last layers are too complex,



(a) GPU and CPU profiling

(b) GPU to CPU speedup

(c) Hybrid solution

(d) Data transfer bandwidth

Fig. 6. Experimental results

and require up to 30 scans per layer (layer 11), so the GPU spends more time classifying the frame.

Because of the high data parallelism support, the GPU is running faster than the CPU for the first 6 layers as shown in Fig. 6(b). On the other hand, the next 6 layers are running faster on the CPU, because the small number of windows to classify and the high number of features within the layers.

Fig. 6(c) presents the profiling of the hybrid solution: the first 6 layers are running on the GPU and the next 6 layers are running on the CPU. Using this decomposition, the face detection reachs a real-time classification with 15 fps for 415x255 frames.

6 Conclusion

We have presented a real-time implementation of an efficient object detection system on the graphics hardware. We have designed our application using the streaming model of computation.

To the best of our knowledge, this is the first attempt to implement a sliding-window technique for visual object-detection on GPU, and the results are very promessing.

References

1. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: European Conference on Computational Learning Theory. (1995) 23–37
2. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 23–38
3. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA (1995)
4. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision (2002)
5. Abramson, Y., Steux, B., Ghorayeb, H.: Yef real-time object detection. In: ALART'05:International workshop on Automatic Learning and Real-Time. (2005) 5–13
6. Viola, P.A., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision **63** (2005) 153–161
7. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Proc. IEEE International Conf. Image Processing. Volume 1. (2002) 900–903
8. Abramson, Y., Steux, B.: Hardware-friendly pedestrian detection and impact prediction. In: IVS04. (2004) 590–595
9. Fung, J., Mann, S.: Using multiple graphics cards as a general purpose parallel computer: applications to computer vision. In: ICPR 2004. Proceedings of the 17th International Conference on Pattern Recognition. Volume 1. (2004) 805 – 808
10. Fernando, R., Kilgard, M.J.: The Cg Tutorial. first edn. (2003)
11. Buck, I., Hanrahan, P.: Data parallel computation on graphics hardware. Graphics Hardware (2003)
12. Venkatasubramanian, S.: The graphics card as a stream computer. Workshop on the Management and Processing of Data Streams (2003) AT&T Labs Research.
13. Buck, I., Foley, T., Horn, D., Sugerman, J., Mike, K., Pat, H.: Brook for gpus: Stream computing on graphics hardware. ACM Transactions on Graphics (2004)
14. Segal, M., Akeley, K.: The OpenGL graphics System: A specification(Version 2.0). (2004)

Combining Iterative Inverse Filter with Shock Filter for Baggage Inspection Image Deblurring

Guoqiang Yu[1], Jin Zhang[2], Li Zhang[3], Zhiqiang Chen[3], and Yuanjing Li[3]

[1] Department of Application, Nuctech Company Limited, 100084, P.R. China
[2] Department of Automation, Tsinghua University, 100084, P.R. China
[3] Department of Engineering Physics, Tsinghua University, 100084, P.R. China

Abstract. In this paper, we describe an image deblurring algorithm for images generated by the baggage inspection system. Baggage inspection images have low-extent blurring, large intensity dependent noise and need line by line processing in real time, which makes most of the existing methods unsuitable. With these special characteristics, we propose a new algorithm by combining the iterative inverse filter and the shock filter. At each iteration of the inverse filter, the constraint borrowed from the shock filter is imposed so that the image is deblurred without ringing artifacts. The algorithm is fairly fast and can process the image line by line, which can satisfy the real-time requirement. It is also easy to program and can be implemented in practice. The algorithm is tested on the synthetic data and real data from the airport. The experiments show that our algorithm has a great improvement on human's perception and is better than the original algorithms.

1 Introduction

The detection of explosives and contraband goods is important to fight against terrorism and smuggling. Alerted by security failures in the recent past, the practice and research in advanced scanning equipments and associated technologies have become a priority. For example, beginning on January 1, 2003, all baggage from U.S. transported by air are screened for explosives/explosive devices. There are many kinds of methods for detection of explosives and contraband goods, including X-ray based screening, nuclear based explosives detection, electromagnetic detection and vapor detection[13]. The baggage inspection system is an X-ray based scanning system that provides safe and non-intrusive security solutions.

The baggage inspection system is a translation-scanning system with a line-detector array. An X-ray generator with collimator forms a slice X-ray source. The X-ray slice penetrates baggages and a line of image pixels is obtained through detectors and signal collection subsystem. The baggages are carried by a driving equipment and scanned line by line, so a digital projection image is formed. The structure of a typical baggage inspection system is shown in Fig.1.

In a baggage inspection system, the equipment measures how many X-ray photons are attenuated from the illuminating beam at each location of the bag. Due to the photoelectric effect and the Compton effect, when a beam of incident

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 264–273, 2006.

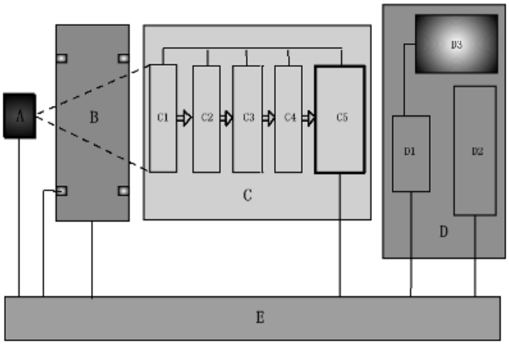


Fig. 1. Structure of a typical baggage inspection system. A is the X-ray generator; B is the driving device; C is detectors and signal collection subsystem, including the detectors(C1), the preamplifier(C2), the main amplifier(C3), A/D converter(C4) and the control and communication module (C5); D is the running check subsystem, which is composed of the host computer(D1), the control board(D2) and the monitor(D3); E is the electrical control subsystem.

X-rays traverses through any material, its intensity is reduced[5]. Suppose the incident radiation has an intensity of I_0, the average output intensity I after passing some material is,

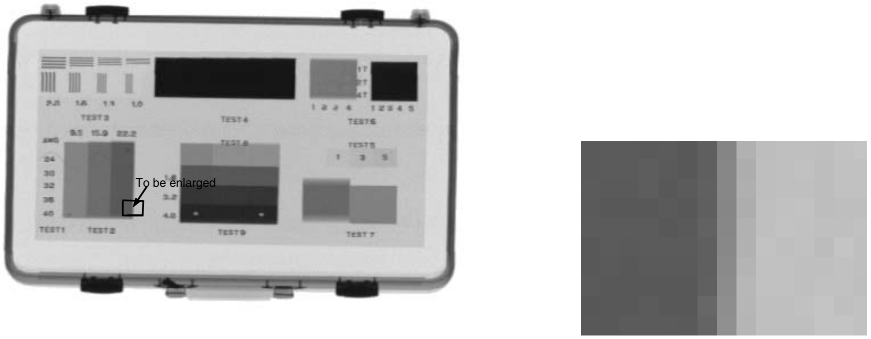
$$I = I_0 e^{-\sigma n x} \tag{1}$$

where x is the thickness of the material, n is the number of the atoms per unit volume and σ is cross-section per atom. When the material is placed in the X-ray radiation field, the intensity of the emergent X-ray at different location lies on the property and the thickness of the material that the X-ray penetrates through, so the emergent X-ray intensity can reflect the distribution of the material.

The image generated by the baggage inspection system has its special characteristics. Firstly, the noise is large and intensity dependent. The number of the X-ray photon passed through obeys the Poisson distribution[5]. According to the property of the Poisson distribution when the X-ray intensity is very large its variance can be ignored comparing to the intensity itself. Unfortunately, considering the radiation safety, the baggage inspection system can't adopt large intensity X-ray, so the noise inherited in the Poisson variance can't be ignored. In general, the noise to signal ratio of the baggage inspection image can reach 2% at medium gray level. The signal collection circuits can also cause some noise, which is usually very small and can be ignored.

Secondly, the baggage inspection image has low-extent blurring. To be sensitive to the variance of the matter's thickness, the size of the detector can't be small and the image is collected by moving the matter with a not very small speed, so the image has some blurring. For a typical baggage inspection system, each pixel in the image corresponds to about 1.35mm along motion direction and the size of the detector along this direction is 2.7mm, so the image has blur of about 2 pixels. Though the blurring is not big, it affects the perception to the observer and automated judgement greatly.

To give an intuitive impression, Fig.2(a) shows a typical image from a baggage inspection system and Fig.2(b) shows an enlarged area in the region of Fig.2(a) as pointed out. From Fig.2(a) we can see that the words in the image such as 'TEST1' and 'TEST2' are not distinct and the edge is smeared. The enlarged image indicates the blurring and noise clearly. The image generated from the baggage inspection system is the basis for the operator to judge whether the bag contains some explosive or illicit materials. A blurred image is not appropriate for this purpose. The blurring can weaken the operator's acuity greatly and fatigue the operator's eyes. For automated judgement, the deblurring is also a necessary preprocessing step. So in this paper we devote ourselves to the research of the deblurring algorithm for the baggage inspection image.



(a)

(b)

Fig. 2. (a)A typical image from baggage inspection system; (b)An enlarged area of Fig.2(a)

The remainder of this paper is organized as follows. In section 2 we describe some previous algorithms for image deblurring and point out their advantages and disadvantages for our baggage inspection image. In section 3 we combine the iterative inverse filter and the shock filter and propose our new algorithm. Our new algorithm is described in details and the algorithm schedule is also presented. In section 4 we demonstrate two experiments, one for synthetic data and one for real world data from the airport. The last section concludes the paper and points out some prospective research directions.

2 Previous Work

Deblurring is a classical and hard problem in image processing community. The original image $f(x, y)$ is blurred by a point spread function (PSF) $h(x, y)$ with contamination by the noise. If we denote $g(x, y)$ as the observed image (the blurred image), the procedure can be described as

$$g(x, y) = h(x, y) \otimes f(x, y) + n(x, y) \tag{2}$$

where $\otimes$ represents the space convolution.

The objective of deblurring is to find the best estimation of the original image given the observed image and knowledge of the PSF and the noise. There are many algorithms to tackle the blurring, such as (iterative) inverse filter[7, 8], Wiener filter and constrained least square filter[2], Lucy-Richardson method[8]. Other methods, including TV preserving image restoration and the shock filter[11][1], are also proposed. These algorithms are deduced from different aspects and can deal with different cases. Considering that our baggage inspection image needs line by line processing, we only review two methods:the iterative inverse filter and the shock filter.

2.1 Iterative Inverse Filter

The inverse filter is very popular for image deblurring for its elegant representation. But not all the PSF can be inverted and even when the PSF is really invertible, it will cause noise amplification in the restored image, because the image degradation is usually a low-pass process. As a compromise, the inverse filter is usually implemented by iterative method.

Use $\hat{f}(x, y)$ to denote the estimation of the original image, ignoring the noise, it should hold for all parameters β,

$$\hat{f}(x, y) = \hat{f}(x, y) + \beta(g(x, y) - h(x, y) \otimes \hat{f}(x, y)) \tag{3}$$

Applying the method of successive substitution to Eq.3 suggests the following iteration scheme:

$$\begin{aligned} \hat{f}_0(x, y) &= g(x, y) \\ \hat{f}_{k+1}(x, y) &= \hat{f}_k(x, y) + \beta(g(x, y) - h(x, y) \otimes \hat{f}_k(x, y)) \end{aligned} \tag{4}$$

If there is an solution for the inverse filter and the parameter β is not too large, Eq.4 can converge to the solution of the inverse filter. Moreover, the noise amplification effect can be minimized by terminating the algorithms after a finite number of iteration[8].

The iterative method in essence, is one kind of regularization method[8], hence it shares the limitations of many regularization methods, for example the recovery quality of the image is decreased, ringing occurs around the prominent features of the restored image, *etc.* Many algorithms are proposed to improve the iterative inverse filter. Typically some priori information about the original image is incorporated in the iterative procedure. These priori information include the space structure of the original image[12], assuming the original image to be nonnegative and with finite support [4] and the HVS[9] to name a few. However, as the constraints in these methods are usually of quadratic form, they can't remove ringing completely. The computational requirement of these methods also limits their usage in real time applications.

2.2 Shock Filter

As in our baggage inspection image the blurring is just 2 pixels, other methods, belonging to image enhancement, not conventional image restoration, are also applicable. Shock filter is such an algorithm.

The shock filter is based on nonlinear time dependent partial differential equations[6][14]. Consider a continuous image $f : \mathcal{R}^2 \to \mathcal{R}$, a class of shock filtered image $\{u(x, y, t)|t \geq 0\}$ of $f(x, y)$ may be created by evolving f under the process:

$$u_t = -sign(\triangle u)|\nabla u| \tag{5}$$

with the initial value

$$u(x, y, 0) = f(x, y) \tag{6}$$

Here, subscripts denote partial derivatives, and $\nabla u = (u_x, u_y)^T$ is the (spatial) gradient of u.

When implementing the shock filter, the above equations should be discretized. Osher and Rudin have developed a scheme to preserve the variation and the size and location of the extrema[10]. In their scheme, u_x, u_y are approximated by

$$u_x = m(\Delta^x_+, \Delta^x_-) \tag{7}$$

$$u_y = m(\Delta^y_+, \Delta^y_-) \tag{8}$$

$$\Delta^x_\pm u = \pm(u(x \pm 1, y) - u(x, y)) \tag{9}$$

$$\Delta^y_\pm u = \pm(u(x, y \pm 1) - u(x, y)) \tag{10}$$

and $m(x, y)$ is the minmod function defined by

$$m(x, y) = \begin{cases} (signx)\ \min(|x|, |y|) \qquad if \quad xy > 0 \\ 0 \quad if \quad xy \leq 0 \end{cases} \tag{11}$$

The main properties are:

- Shocks develop at inflection point(second derivative zero-crossings) and within a region the image is smoothed;
- Local extrema remain unchanged in the procedure. No new local extrema are created;
- The steady state solution is piecewise constant;
- The process approximates deconvolution.

However, the shock filter doesn't take the degradation model into account. For the original image without blurring the shock filter will also generate a shock at the inflection points. As to the deblur property, shock filter can only deal with the blurring of the step edge. When there are some impulse profiles besides the step edge in an image, using shock filter solely can't recover the impulse profiles.

3 Our Algorithm

Our baggage inspection image is a low-extent blurred image with relatively large noise and needs to be processed line by line in real time. From the above analysis, we can see that both the iterative inverse filter and the shock filter are unsuitable.

The iterative inverse filter makes a best fit of the observed image and can be implemented line by line in real time. However, although the amplification of the noise can be suppressed by terminating the algorithm after a finite number of iterations, ringing around the edges of the image and mosaics in the smooth region occur.

On the other hand, the shock filter satisfies maximum principle and generates no new extremum, i.e, no ringing occurs, which is a desirable property. However, shock filter uses only the information of the image itself. Shocks are developed at the inflections points of the image. Without other guiding information, fake shocks will be developed when processing line type feature or impulse type features. As we can see in Fig.2, there are many line type features to be processed in our baggage inspection image, for example, the fuse in the detonator, which is an important clue to judge whether there are some explosives.

A simple idea to remedy these problems is to combine these two algorithms to propose a new algorithm for the baggage inspection image, that is, we adopt the iterative inverse filter as the basic algorithm with some constraints inspired by the shock filter.

Notice that ringing is the new extremum generating by the iterative inverse filter. They are around the extremum of the original image. To alleviate ringing, it is best to consider the image in bounded variation space as the shock filter does. In the shock filter, to guarantee that no extremum will generate, the amplitude of the change at any point should be not more than the amplitude of the differences between the current point and its adjacent points. Moreover, to preserve the total variance of the image the shock filter makes the value of the point unchanged when the differences of the point and its adjacent points are with different signs. To eliminate the oscillatory phenomena in the iterative inverse filter, we can also put the similar constraints on the update of the image.

Denote the difference between the successive iterations $t-1$ and t as I_t,

$$I_t(x,y) = \hat{f}_t(x,y) - \hat{f}_{t-1}(x,y) \tag{12}$$

From Eq.4 we can get I_t^{iif} for the iterative inverse filter,

$$I_t^{iif}(x,y) = \beta(g(x,y) - h(x,y) \otimes \hat{f}_k(x,y)) \tag{13}$$

Assume the baggage moves along the x direction. To eliminate the oscillatory phenomena the amplification of I_t should not be more than the amplification of I_t^{mx} or I_t^{px},

$$I_t^{mx} = \hat{f}_t(x,y) - \hat{f}_t(x-1,y) \tag{14}$$

$$I_t^{px} = \hat{f}_t(x+1,y) - \hat{f}_t(x,y) \tag{15}$$

The minmod function in shock filter guarantees that the local extremum of the original image remain unchanged, that is, the extremum of the processed image is just the extremum of the observed image. This property, obviously is not desired for image deblurring application. So we cancel it in our algorithm. As a result, we set up I_t as

$$I_t = sign(I_t^{iif})\ \min(|I_t^{mx}|, |I_t^{px}|, |I_t^{iff}|) \tag{16}$$

To sum up, we present the algorithm procedure as in Table(1).

Table 1. The schedule of our proposed algorithm

Step 0: *Initialization*

- Set up the number of iteration *iter*
- Set up the updated coefficient β
- Set up the initial estimation $\hat{f}_0(x, y) = g(x, y)$
- Let $t = 1$

Step 1: *Calculate the difference* I_t

- Calculate I_t^{mx} and I_t^{px} for each point according to Eq.14 and 15
- Calculate I_t^{iff} for each point according to Eq.13
- Calculate the difference I_t for each point according to Eq.16

Step 2: *Update the image*

- Let $\hat{f}_t(x, y) = \hat{f}_{t-1}(x, y) + I_t$
- Let $t \leftarrow t + 1$
- Compare t and $iter$. If $t > iter$, stop the iteration, otherwise, go to **Step 1**

The proposed algorithm can be examined from two viewpoints. First, from the iterative inverse filter's point of view, new constraint borrowed from the shock filter has been added to the iteration procedure. This constraint can guarantee that no new extremum will produce so the disturbing ringing effect in conventional methods will not occur. And this TV constraint is added in a way very different with the way proposed in [3].

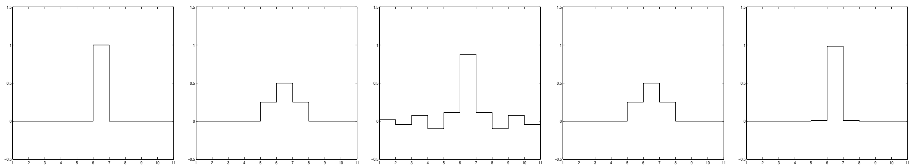
Second, we can think the proposed algorithm from the shock filter's point of view. In the shock filter, shocks are developed according to the properties of observed image itself. No other information is added. The proposed algorithm incorporates the image degradation model into the shock filter. Shocks are developed under the guidance of the inverse filter. As a result, it can deblur image that shock filter can't do, such as impulse signal, staircase signal, *etc.*

The proposed algorithm is suitable to the baggage inspection image. As stated previously, the baggage inspection image has only a low-extent blurring. The constraint borrowed from shock filter is valid. Moreover, the algorithm is fairly fast and can process the image line by line, which can satisfy the real-time requirement. It is also easy to program and can be implemented in practice.

4 Experimental Results

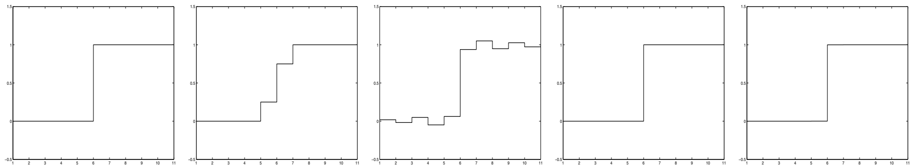
4.1 One-Dimensional Toy Data

In this experiment we will show how our proposed algorithm performs on the blurred step and impulse signal. The performance of the iterative inverse filter and the shock filter is also presented.



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

Fig. 3. Experiment for the impulse signal. (a)Original signal; (b)The blurred signal; (c) The deblurred signal with the iterative inverse filter; (d)The deblurred signal with the shock filter; (e) The deblurred signal with our proposed method.



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

Fig. 4. Experiment for the step signal. (a)Original signal; (b)The blurred signal; (c) The deblurred signal with the iterative inverse filter; (d)The deblurred signal with the shock filter; (e) The deblurred signal with our proposed method.

Fig.3 and Fig.4 show the results of the experiments for the impulse signal and the step signal respectively. The subfigure (a) in these two figures shows the original signals. A linear motion blurring is applied to the original signals, resulting in the blurred signal as in subfigure (b). We apply the iterative inverse filter to deblur the signal in Fig.3(b) and Fig.4(b). The deblured signals are shown in Fig.3(c) and Fig.4(c). The iteration number is set to 10. Just as we anticipated, both the deblurred impulse and step signal have oscillatory phenomena.

Fig.3(d) and Fig.4(d) are the deblurred signals with the shock filter. Since the shock filter satisfies a maximum principle and generates no new extremum, there are no oscillatory phenomena. However, it is not effective for the impulse signal. In addition, if the original signal is just like the signal in the subfigure (b) of Fig.4. the shock filter will also process it to a step (shock), which is not desired.

We use our proposed algorithm to process the signals and get the results as the Fig.3(e) and Fig.4(e), which indicate that our algorithm has the advantages of both the iterative inverse filter and the shock filter. Our proposed method has no oscillatory phenomena and it can deal with the impulse signal.

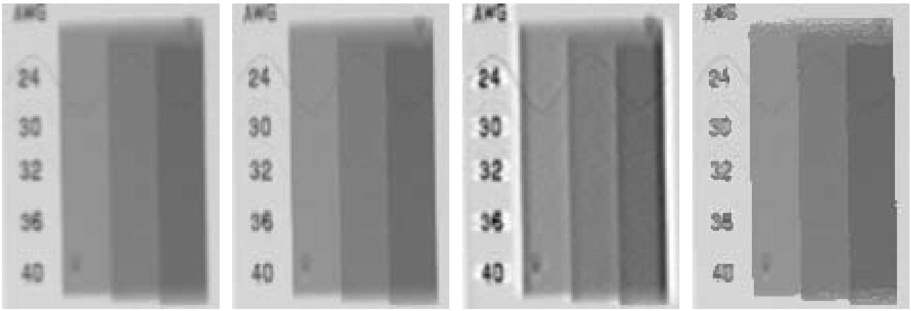
4.2 Real-World Data

Our proposed algorithm is also tested on the real-world data from an airport and compared to other four algorithms, the iterative inverse filter, the shock filter, the Wiener filter and the inverse filter. As stated in section 1, the baggage inspection system used in this experiment carries the baggage at a speed of 0.2m/s and collects the data at a frequency of 150Hz, so a pixel in the image corresponds to 1.33mm. The detector unit is 2.7mm long in the direction of motion. So the blurring is about 2 pixels, thus the PSF can be written as

$$h(x,y) = \begin{cases} 0.25 & x=-1, y=0 \\ 0.5 & x=0, y=0 \\ 0.25 & x=1, y=0 \\ 0 & else \end{cases} \tag{17}$$

where x denote the motion direction.

To compare the results clearly, we use an enlarged figure as shown in Fig.5. From these four subfigures we can see that our proposed algorithm has the most impressive result. The result of the iterative inverse filter has serious ring artifacts. The result of the shock filter is distortional since it can't consider the degradation model.



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

Fig. 5. The experiment result for the real-world baggage inspection image. (a) A part of the observed baggage inspection image; (b) the deblurred image with our proposed algorithm; (c) the deblurred image with the iterative inverse filter, which has serious ring artifacts; (d) the deblurred image with the shock filter, which is distortional.

5 Conclusion and Discussion

In this paper, we study the problem of baggage inspection image deblurring. We first introduce the baggage inspection system and explain how it works. The characteristics of the baggage inspection image are depicted. The image has a low-extent blurring with relatively large noise. It is necessary to deblur the image.

There are many deblurring algorithms and we review some of the classical algorithms. All these algorithms have some disadvantages and are not suitable to our application. We combine the iterative inverse filter and shock filter to propose a new algorithm. At each iteration of the inverse filter, constraint borrowed from the shock filter is added so that the image is deblurring without ringing effect.

The new proposed algorithm is tested on the synthetic one-dimensional signal and the real-world data. We also compare it to other algorithms on these data. The experimental results indicate that our algorithm is effective and can be applied in practice.

We expect this paper can intrigue some researchers in the signal processing, image analysis and pattern recognition communities. There are many problems in the processing of the baggage inspection image. The segmentation of the image and automatic alarming are still active research directions for the baggage inspection system. There are some new systems such as the dual energy system. How to fuse the different energy information is also an interesting topic.

References

1. L. Alvarez and L. Mazorra. Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal on Numerical Analysis*, 31(2):590–605, Apr. 1994.
2. H. C. Andrews and B. R. Hunt. *Digital Image Restoration.* Prentice-Hall, 1977.
3. P.L. Combettes and J.C. Pesquet. Image restoration subject to a total variation constraint. *IEEE Trans.Image Processing*, 13(9):1213–1222, Sept. 2004.
4. D.Kundur and D. Hatzinakos. A novel blind deconvolution scheme for image restoration using recursive filtering. *IEEE Trans. Signal Processing*, 26(2):375–390, Feb 1998.
5. N. A. Dyson. *X-rays in atomic and nuclear physics.* Longman, second edition, 1990.
6. Guy Gilboa. *Super-Resolution Algorithms Based on Invers Diffusion-type Processes.* PhD thesis, Israel Institute of Technology, 2004.
7. Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Pearson Education, Inc., second edition, 2002.
8. R.L. Lagendijk J. Biemond and R.M. Mersereau. Iterative methods for image deblurring. *Proceedings of the IEEE*, pages 856–883, May 1990.
9. A. K. Katsaggelos and S. N. Efstratiadis. A class of iterative signal restoration algorithms. *IEEE Trans. Accus. Speech and Signal Processing*, 38:778–786, 1990.
10. Stanley Osher and Leonid I. Rudin. Feature-oriented image enhancement with shock filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, Aug 1990.
11. Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Matching Intelligence*, 12(7):629–639, July 1990.
12. R.L.Lagendijk, J. Biemond, and D.E. Boekee. Regularized iterative restoration with ringing reduction. *IEEE Trans. Accus. Speech and Signal Processing*, 36(12):1874–1888, Dec 1988.
13. Maneesha Singh Sameer Singh. Explosives detection systems (eds) for aviation security. *Signal Processing*, (83):31–55, 2003.
14. Guillermo Sapiro. *Geometric partial differential equations and image analysis.* Cambridge University Press, 2001.

Automatic Chromosome Classification Using Medial Axis Approximation and Band Profile Similarity*

Jau Hong Kao[1], Jen Hui Chuang[2], and Tsai Pei Wang[2]

[1] Department of Computer Science, National Chiao Tung University,
Hsinchu City, 30010, Taiwan
gis88804@cis.nctu.edu.tw

[2] Department of Computer Science, National Chiao Tung University,
Hsinchu City, 30010, Taiwan
{jchuang, wangts}@cs.nctu.edu.tw

Abstract. Automated chromosome classification is an essential task in cytogenetics and has been an important pattern recognition problem. Numerous attempts were made in the past to characterize chromosome to perform clinical and cancer cytogenetics research. It is important to determine optimum features and develop feature extraction schemes for chromosome classification. In this paper we propose novel approaches for medial axis determination and profile matching of human chromosome. The medial axis determination plays a critical role for precise and complete extraction of band patterns of chromosomes. The features of the band profile obtained along the axis are then used in the classification process. In particular, the medial axis is obtained by simple cross-section analysis and the classification is accomplished through iterative pairing of band profiles. According to the experimental results, the developed automatic system can efficiently extract band profiles of the chromosomes along their medial axes, and satisfactory chromosome classification results can be obtained.

1 Introduction

Chromosome classification is an essential task in cytogenetics and has been an important pattern recognition problem. Numerous attempts were made in the past to characterize chromosome for prenatal screening and genetic syndrome diagnosis, cancer pathology research and environmentally induced mutagen dosimetry. However, chromosome classification and analysis, which create karyotypes, are manually performed in most cytogenetics laboratories nowadays in a repetitive, time-consuming and therefore expensive procedure [1]. Hence, the development of computerized methods to automate the procedure has attracted much attention. It is important to determine optimum features and develop feature description scheme for chromosome classification. In general, the features used in the classification include shape description and band pattern representation.

* This work is partly supported by Lumens Digital Optics, Inc. and by Ministry of Economic Affairs, Taiwan, under grant no. 94-EC-17-A-02-S1-032.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 274–283, 2006.

The band patterns produced by modern specimen staining techniques enable discrimination of all human chromosomes into their corresponding ISCN categories [2]. Many attempts were made to characterize the band patterns as part of the chromosome feature description vector. To perform the classification successfully, precise and complete band pattern of chromosome plays an important role. Banding techniques in [3] and [4] are proposed to classify chromosomes. However, they may not be accurate enough to achieve desired success rate in the classification. On the other hand, shape-based techniques are also proposed. In [5], the curvature of each chromosome is studied and feature vector is extracted from the curvature to recognize the biological class of each chromosome. In [6] and [7], the authors proposed a new way to obtain the chromosome longitudinal axis, or medial axis, based on the dominant points of the contour and cubic splines of the boundary, and used the simple constrained classifier [6] to classify chromosomes. The uses of wavelet packet transform and Fourier transform shape representation also motivate the developments of the approaches for chromosome classification [8]. The performance of shape based approaches depends heavily on the quality of the input images, as well as the resolution of the boundary of the segmented chromosome.

As stated in [1] and [9], the combination of both features from band pattern and shape representation further reduces the error rate of karyotyping systems. It is also desirable to choose features as resistant as possible to different transformations and image noise since the appearance of a chromosome is rarely in good conditions. Medial axis determination, which provides a reliable basis for efficient band profile computation, becomes a required and particularly critical step to obtain most features of chromosomes. However, the calculation of medial axis is difficult [6]. Many approaches are proposed to perform this task [3, 6, 10]. The method in [3] relies on a skeletonisation algorithm to compute the medial axis. Some medial axis calculation methods are based on second order moments of the chromosome grey level values, but they are not applicable for bent chromosomes. In [11], a software tool was developed to straighten curvilinear chromosome after a manual ends-identification process, making the medial axis extraction a straightforward procedure.

In this paper, we proposed several novel algorithms for medial axis determination and chromosome classification through iterative paring of band profiles. The database of 96 DPI chromosome images used in our experiments are provided by Cheng Gung Memorial Hospital and Cathay General Hospital. The dimension of the images is less than 40 × 132 when oriented vertically. The segmentation process is done by Lumens Digital Optics Inc., Taiwan. Since the positions of centromeres of chromosome images in the database are generally difficult to identify using only contour information, the proposed approach does not introduce centromere features for classification. The medial axis is obtained by simple cross-section analysis along four scan-line orientations, as described in Section 2. The proposed band profile extraction scheme along the medial axis is introduced in Section 3. Section 4 gives a description of the procedure for band profile matching, which is an extension of a substring matching algorithm.

Section 5 presents the experimental results and Section 6 gives the conclusion of the paper.

2 Medial Axis Extraction

A major challenge for the automatic classification of a chromosome, or the composition of a karyotype, from a chromosome image, results from the fact that chromosomes can be very sinuous, especially for the classes of long lengths. Therefore, detailed analysis becomes a difficult and tedious task. Fig. 1 shows a segmented binary image of a long, bent chromosome. In general, a practical way to represent elongated objects is by using their longitudinal axes, or medial axes. For a symmetric object such as a chromosome, the lengths of cross-sections perpendicular to the medial axis are likely very close to the nearly constant width of the object. Moreover, the midpoints of such cross-sections will mostly lie close to the medial axis. Given sufficient number of cross-sections, the polygonal curve computed by connecting the midpoints in proper order provides an approximation of the medial axis. In this paper, we propose a method for this process through a simple and efficient cross-section analysis. In order to catch the orientation and the shape of the curvilinear chromosome by cross-sections, we apply scan-lines in the image along the following four orientations: 0°, 45°, 90°, and 135°. By appropriately selecting the cross-sections according to their lengths, and by connecting the midpoints of these cross-sections in the correct order, we can obtain the approximation of the medial axis of the object.

The cross-section analysis mainly performs separation and merging of line segments which are connections of groups of midpoints of cross-sections. They are vital steps to successfully compute the final polygonal curve as the approximation of the medial axis. The separation includes filtering and grouping processes, with the objectives being to sort out incorrect or improper middle points and to organize the remaining ones into line segments according to the shape structure of chromosome, respectively. The objective of merging is to derive the polygonal curve by properly connecting the obtained line segments.

In the separation step, a filter is defined based on an analysis of lengths of cross-sections. The representative midpoints for chromosomes are in fact the subset that the lengths of the corresponding cross-sections are within a specific range relative to the width of the chromosome. The range can be determined relatively by statistical analysis on the lengths of cross-sections. We first calculate

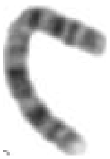
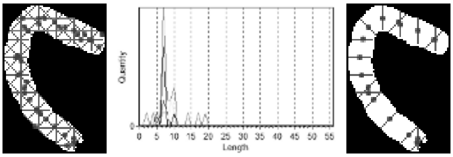


Fig. 1. An image of a chromosome

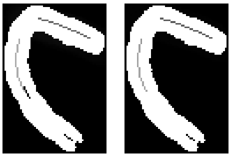


(a) (b) (c)

Fig. 2. Statistical analysis of lengths of cross-sections of the chromosome in Fig. 1. (a) All cross-sections. (b) The quantities of different lengths of cross-sections along different orientations. (c) The result after filtering the cross-sections.

the histograms of cross-section lengths. Fig. 2(a) shows the cross-sections of the chromosome shown in Fig. 1. Fig. 2(b) shows the length histograms separately for cross-sections along different orientations. One can find that the four peaks of the distributions occur at nearly the same length, e.g., the range from 5 to 10 in the figure. In fact, the representative range of length fairly corresponds to the width of the chromosome of interest. The range is thus used to define the filter for the cross-sections. The out-of-range ones, e.g., the longest vertical cross-sections shown in Fig. 2(a), can be discarded this way. Note that because the filter is a relative measurement to the shape and to the scale of individual chromosome image, it is adaptive to databases. Fig. 2(c) shows the selected cross-sections.

The grouping process in the separation step aims to organize midpoints of filtered cross-sections according to the shape of the chromosome. The medial axis of a chromosome is a simple curve and can be approximated by a series of line segments. We use consecutive cross-sections with same orientation to compute these segments. Specifically, we start by connecting the midpoints of consecutive cross sections of the same orientations. Fig. 3(a) shows the four pieces of polygonal curves, each consisting of connected line segments, obtained from cross-sections of three different orientations represented by three grey levels. Note that the line segments should not cross each other or the boundary of chromosome. However, such problems do occur at times. For example, two pieces of line segments do cross each other in Fig. 3(a). A process based on simple split and gradual shortening algorithm is introduced to simplify the connecting process for the segments. For each pair of crossed line segments, the shorter one is further shortened by deleting the end point nearby the intersec-



(a) (b)

Fig. 3. (a)Line segments obtained after grouping process. (b)The simplified line segments.

tion till the two line segments are isolated. A similar process is proceed for line segments that cross the boundary of the chromosome. The touched and overlapped line segments are subsequently adjusted into disconnected ones, as shown in Fig. 3(b).

The merging algorithm is an iterative process. First, the line segments obtained previously are connected progressively till there is only one single polygonal curve. A pair of line segments having the most closed end points is connected in each iteration without crossing or touching the boundary of the chromosome. It is desirable that the final polygonal curve is as long as possible. Hence, a threshold λ is defined as certain percentage of the length of the longer side of image as an approximation of chromosome length. If the length of the final polygonal curve is less than λ, we return to the cross-section selection stage and increase the range of allowed cross-section length by 10% of the maximum cross-section length in order to select more cross-sections, which in turn allow us to extract possibly longer curves by connecting their midpoints. Once we have obtained a polygonal curve with length of at least λ, it is extended by finding the tip points on the chromosome boundary and connecting the end points of curve to them, respectively. To precisely determine the tip points, the grey image of chromosome is used and a search area is defined on the boundary nearby the end points. We then extend the polygonal curve from its end points along the direction perpendicular to the bands, which is found by searching the chord having minimum of normalized variance of intensity. Note that, similarly, if the end point is too far away from the tip point, i.e., longer than two times of the filter range which is an approximate of the chromosome width, the range of filter will be relaxed and the system will performs cross-section analysis and medial axis extraction again. Fig. 4 shows the polygonal curves of two chromosomes after the extension process along with their corresponding band images. We can see that these curves are good approximations of the medial axes.

In the final step, we perform a smoothing procedure for the final polygonal curve to determine precise medial axis. We first perform linear interpolation of slope along the extended polygonal curve. For each sample point between two adjacent vertices of the polygonal curve, its slope is defined as a weighted linear combination of slopes of the line connecting the two vertices and the adjacent line segments. The polygonal curve is thus transformed into a continuous and smooth

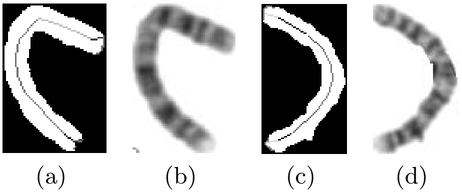
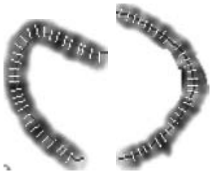


Fig. 4. (a), (c) Two results after connecting end points of polygonal curves to tip points of boundaries of chromosomes. (b), (d) The corresponding band images of (a) and (b), respectively.

one. Finally, new cross-sections of a chromosome are computed at particular intervals by intersecting the lines perpendicular to the smoothed curve and the chromosome boundary. The midpoints of these cross-sections are then connected sequentially to form the precise medial axis of the chromosome for subsequent extraction of its band profile.

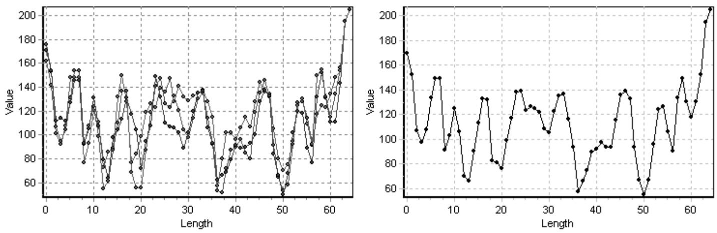
3 Band Profile Extraction

The medial axis obtained previously is the basis for the extraction of the band profile of a chromosome. The grey values of band profile are sampled at pixels on the cross-sections perpendicular to the medial axis. To avoid the noisy data at the boundary pixels of chromosome, and to reduce outliers mainly caused by the bending of chromosomes and during the image acquiring process, we compute the profile using weighted combination of grey values sampled at 1/4, 1/2, and 3/4 of width along each cross-section. Fig. 5 shows the cross-sections used for this purpose for two chromosomes. Fig. 6(a) presents the three curves, each consisting of grey values obtained at one of the three points (1/4, 1/2 and 3/4 width) on the cross-sections in Fig. 5(a). Fig. 6(b) shows the band profile of the chromosome obtained by combining the three curves in Fig. 6(a) with a weighted sum computation. It is obvious that the outliers of grey values presented in Fig. 6(a) are significantly reduced through the voting process.



(a) (b)

Fig. 5. (a) An example of the cross-sections used to extract band profiles. (b) Another example.



(a) (b)

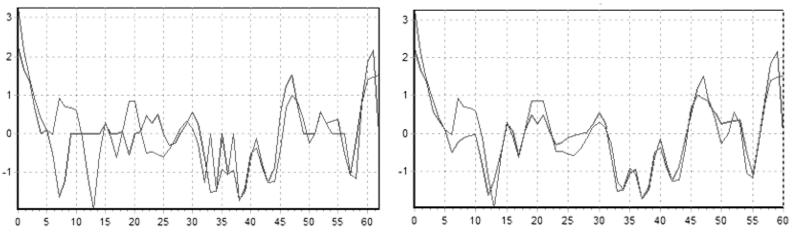
Fig. 6. (a) The three curves of grey values obtained along the medial axis at the 1/4, 1/2 and 3/4 of width of cross-section of the chromosome in Fig. 5(a). (b) The band profile extracted by combining the three curves of (a).

4 The Matching Algorithm Using Band Profile Similarity

To successfully match chromosomes, we developed an iterative pairing approach which matches band profiles using local and global similarities. Two kinds of normalization are taken into account for the matching process. The grey values of profiles are normalized by Z-score computation. Since the alignment process eventually adjusts the two band profiles to the same length, the normalization of length of band profiles is basically not necessary.

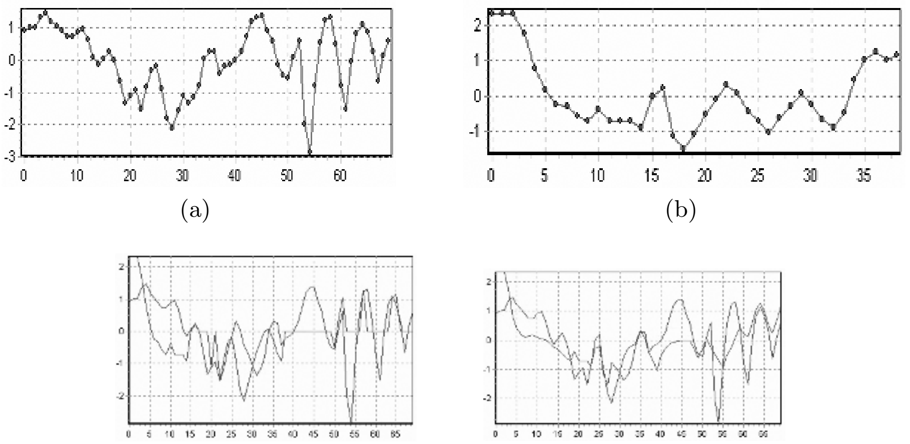
The matching of two band profiles under consideration is based on the basic substring matching algorithm using dynamic programming technique and thus has the advantage that the problem due to the variance of profile length can be solved through the insertion operation of substring matching algorithm. In contrast to basic substring algorithm, the grey value is not quantized in the proposed approach. The matching algorithm considers both local similarity and global similarity while aligning features on band profiles. Particularly, the adopted local similarity relates to the peakness of the specific feature value along one band profile, and the closeness of the two feature values from the two band profiles. Global similarity is related to the neighboring values. That is, if the alignment of the neighboring values can result higher overall score than the values of interest, then the profiles are likely to be aligned at the neighboring feature values. Fig. 7(a) gives the result of matching of two band profiles using basic substring matching algorithm, and Fig. 7(b) gives the result using the proposed similarity-based approach. One can see from this example that the proposed approach can successfully obtain more satisfactory alignment of band profiles than basic substring matching algorithm.

Fig. 8 demonstrates another example of band profile alignment. The normalized band profiles to be matched are shown in Fig. 8(a) and Fig. 8(b). Note that the lengths of the two band profiles are 70 and 39, respectively. Fig. 8(c) shows the alignment result using basic substring matching algorithm and Fig. 8(d) shows the result using the proposed similarity based algorithm, respectively. Both results have the same length after the band profiles are aligned. Note that in Fig. 8(c) the zeros due to the insertion operation of the substring matching algorithm will produce zero score in the correlation.



(a) (b)

Fig. 7. (a) The matching of band profile using basic substring matching algorithm. (b) The matching of band profile using the proposed algorithm considering local and global similarities.



(a) (b)

(c) (d)

Fig. 8. (a) A band profile of a chromosome. (b) A band profile of another chromosome. (c) The alignment result using basic substring matching algorithm. (d) The alignment result using the proposed method.

The classification of chromosome is accomplished by the developed iterative pairing procedure using above matching algorithm. In the sense of consistency check, if the highest scores are given to each other for two specific chromosomes, the two chromosomes are considered as successfully paired. The above procedure is repeated iteratively until no chromosomes can be paired successfully. Finally, the paired chromosomes are matched to previously obtained templates of band profiles for each class of chromosome to obtain the classification result and the karyotype of chromosomes can be composed eventually.

5 Experimental Results

The data sets used in the experiments are mainly provided by Cheng Gung Memorial Hospital and Cathay General Hospital. The segmentation of the images of chromosomes is carried out by Lumens Digital Optics Incorporation. Fig. 9 shows images of chromosomes of a sample cell.

In our experiments, the medial axis of chromosome can be extracted consistently. Without the curvature computation used in other approaches, the extraction process performs efficiently. The pairing process is a relatively time-consuming stage, mainly because of the heavy correlation computation. However, this can be improved if the size/shape constraint of chromosome is introduced. By grouping chromosomes according to their sizes and areas in advance, most correlation computation can be avoided. Fig. 10(a) shows a matrix of matching scores of band profiles extracted from chromosomes shown in Fig. 9 where the format of the axes is $class\_no - ith\_chromosome$. Fig. 10(b) shows the same matrix as that in Fig. 10(a) with the cells of maximum score of each row labelled

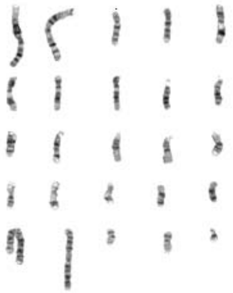
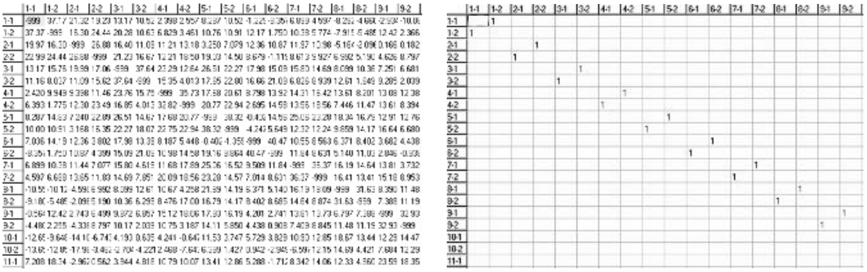


Fig. 9. The images of chromosomes of a sample cell in the data sets used in experiments



	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	6-1	6-2	7-1	7-2	8-1	8-2	9-1	9-2
1-1	-999	37.17	21.32	19.23	13.17	10.52	2.398	2.557	8.237	10.52	-1.225	-8.356	6.899	4.597	-8.292	-4.660	-2.934	-10.06
1-2	37.37	-999	15.30	24.44	20.28	10.63	6.829	3.461	10.76	10.91	12.17	1.750	10.33	5.774	-7.915	-5.485	12.42	2.366
2-1	19.97	16.30	-999	26.88	16.40	11.09	11.21	13.18	3.250	7.079	12.36	10.87	11.97	10.98	-5.164	-2.096	0.166	0.182
2-2	22.99	24.44	25.88	-999	21.23	16.67	12.21	18.50	19.03	14.50	8.679	-1.115	8.613	9.927	6.992	5.190	4.626	8.797
3-1	13.17	15.76	19.99	17.06	-999	37.64	23.29	12.64	26.51	22.27	17.98	15.09	15.80	14.69	8.099	10.36	7.251	6.681
3-2	11.16	8.037	11.09	15.62	37.64	-999	15.35	4.013	17.95	22.80	16.66	21.09	6.826	8.939	12.61	1.549	9.285	2.039
4-1	2.420	9.949	9.398	11.46	23.76	15.75	-999	35.73	17.58	20.61	8.798	13.92	14.31	15.42	13.61	8.201	13.08	12.38
4-2	6.393	1.775	12.30	23.49	16.85	4.013	32.82	-999	20.77	22.94	2.695	14.59	13.56	13.56	7.446	11.47	13.61	8.394
5-1	8.287	14.63	7.240	22.89	26.51	14.67	17.68	20.77	-999	38.32	-0.403	14.55	25.05	23.28	18.34	16.79	12.91	12.76
5-2	10.00	10.91	3.168	15.35	22.27	18.07	22.75	22.94	38.32	-999	-4.246	5.649	12.32	12.24	9.859	14.17	16.64	6.680
6-1	7.006	14.19	12.36	3.802	17.98	13.39	8.187	5.448	-0.402	-1.358	-999	40.47	10.55	8.563	6.371	8.402	3.682	4.438
6-2	-8.356	1.750	10.87	4.399	15.09	21.09	10.98	14.58	19.16	9.864	40.47	-999	11.84	8.631	5.140	11.03	2.846	-0.936
7-1	6.899	10.33	11.44	7.077	15.80	4.619	11.68	17.89	25.06	16.52	9.509	11.84	-999	35.37	16.19	14.64	13.81	3.732
7-2	4.597	6.609	13.65	11.83	14.69	7.051	20.09	18.56	23.28	14.57	7.014	8.631	36.37	-999	16.41	13.41	15.18	8.953
8-1	-10.52	-10.12	-4.59	6.992	8.099	12.61	10.67	4.258	21.39	14.19	6.371	5.140	16.19	19.09	-999	31.63	8.390	11.48
8-2	-9.180	-5.485	-2.096	5.190	10.36	6.295	8.476	17.00	16.79	14.17	8.402	8.685	14.64	8.874	31.63	-999	7.388	11.19
9-1	-0.564	12.42	2.743	6.499	9.972	6.857	15.12	18.06	17.93	16.19	4.201	2.741	13.81	13.73	6.797	7.388	-999	32.93
9-2	-4.480	2.255	-4.338	8.797	10.17	2.039	10.75	3.187	14.11	5.850	4.438	0.908	7.409	8.845	11.48	11.19	32.93	-999
10-1	-12.65	-9.646	-14.10	-6.743	4.193	0.635	4.241	-0.647	11.53	3.747	5.729	3.829	10.90	12.85	18.67	13.44	12.29	14.47
10-2	-13.62	-12.89	-17.98	-3.462	-2.704	-4.221	2.468	-7.645	6.339	1.427	0.942	-2.945	-6.597	12.15	14.69	4.421	7.684	12.29
11-1	7.208	18.34	-2.967	0.562	3.944	4.818	10.79	10.07	13.41	12.86	5.288	-1.712	8.342	14.06	12.33	4.560	23.59	18.35

(a)

	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	5-1	5-2	6-1	6-2	7-1	7-2	8-1	8-2	9-1	9-2
1-1		1																
1-2	1																	
2-1				1														
2-2			1															
3-1						1												
3-2					1													
4-1								1										
4-2							1											
5-1										1								
5-2									1									
6-1												1						
6-2											1							
7-1														1				
7-2													1					
8-1																1		
8-2															1			
9-1																		1
9-2																	1	
10-1																		
10-2																		
11-1																		

(b)

Fig. 10. (a)The matching scores of band profiles extracted from previously classified chromosomes of a cell. (b)The same table wherein the cells of maximum score of each row are labelled by 1.

by 1. From this table, one can see that the proposed approach can successfully produce the correct karyotype for all chromosomes of this human cell. Finally, the experiments are performed using 864 chromosomes from 36 cells. Without the size/shape grouping mechanism mentioned above, an overall classification correctness of 82.7% is achieved.

6 Conclusion

Automated chromosome classification is an essential task in cytogenetics. Numerous attempts were made to characterize chromosome to perform clinical and cancer cytogenetics research. In this paper we propose novel approaches for medial axis extraction and profile matching of human chromosome. In particular, the medial axis is obtained by simple cross-section analysis and the classification is accomplished through iterative pairing process. According to the experimental results, the developed automatic system can efficiently obtain band profiles of the chromosome data set along the extracted medial axis, and generate satisfactory chromosome classification results.

References

1. Lerner, B.: Toward a completely automatic neural- network-based human chromosome analysis. IEEE Tran. Systems Man and Cybernetics **28** (1998) 544–552
2. ISCN: An international system for human cytogenetics nomenclature. in Cytogenetics and Cell Genetics (1985)
3. Piper, J., Granum, E.: On fully automatic feature measurements for banded chromosome classification. Cytometry **10** (1989) 242–255
4. Charters, G.C., Granum, E.: Trainable grey-level models for disentangling overlapping chromosomes. Pattern Recognition **32** (1999) 1335–1349
5. Garcia C.U., Rubio A.B., Perez F.A., Hernandez F.S.: A curvature-based multiresolution automatic karyotyping system. Machine Vision and Applications **14** (2003) 145–156
6. Ritter, G., Schreib, G.: Using dominant points and variants for profile extraction from chromosomes. Pattern Recognition **34** (2001) 923–938
7. Ritter, G., Schreib, G.: Profile and feature extraction from chromosomes. International Conference on Pattern Recognition **2** (2000) 287–290
8. Guimaraes, L.V., Schuck, A., E1bern, A.: Chromosome classification for karyotype composing applying shape representation on wavelet packet transform. In: 25th Annual International Conference of the IEEE EMBS. (2003) 941–943
9. Carothers, A., Piper, J.: Computer-aided classification of human chromosomes: a review. Statistics and Computing **4** (1994) 161–171
10. Groen, C.A., Kate, K., A.W.M. Smeulders, Young, T.: Human chromosome classification based on local band descriptors. Pattern Recognition Letter **9** (1989) 211–222
11. Barrett, S.D., C. R. de Carvalho: A software tool to straighten curved chromosome images. Chromosome Research **11** (2003) 83–90

Object Detection Using a Cascade of 3D Models

Hon-Keat Pong and Tat-Jen Cham

School of Computer Engineering,
Nanyang Technological University, Singapore
hkpong@pmail.ntu.edu.sg,
astjcham@ntu.edu.sg

Abstract. We present an alignment framework for object detection using a hierarchy of 3D polygonal models. One difficulty with alignment methods is that the high-dimensional transformation space makes finding potential candidate states a time-consuming task. This is an important consideration in our approach, as an exhaustive search is applied on a densely-sampled state space in order to avoid local minima and to extract all possible candidates. In our framework, a level-of-detail (LOD) 3D geometric model hierarchy is generated for the target object. Each of this model acts as a classifier to determine which of the discrete states are potential candidates. The classification is done through the estimation of pixel and edge-based mutual information between the 3D model and the image, where the classification speed significantly depends on the LOD and resolution of the image. By combining these models of various LOD into a cascade, we show that search time can be reduced significantly while accuracy is maintained.

1 Introduction

In this paper we address the problem of alignment-based object detection, in which a 3D geometric model is transformed to align with the target object in image. Typically, in finding the set of transformation parameters that best align the model with its image, features from the 3D model are matched to image features by measuring their similarity using an evaluation function. The function values associated with each possible transform form an energy landscape in the parameter space. Most of the existing alignment methods use some directed search techniques to find the optimal transformation, but these usually require initialization near the final solution. In contrast, our proposed algorithm attempts to find a global optimal transformation through exhaustive search, but carried out in a computationally efficient manner.

We present a search method performed using a 3D model hierarchy. These models are decimated versions of a polygonal model of the target object and form a level-of-detail (LOD) hierarchy. The 3D models are loaded on-the-fly at run-time and their images rendered using graphics library, bypassing the need to store 2D multiple-view profiles of these models. Figure 1 illustrates an example LOD hierarchy. We note that models with lower LOD can be evaluated much faster due to reduced number of data points, although with lower accuracy. A

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 284–293, 2006.

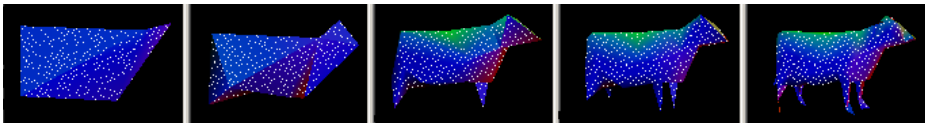


Fig. 1. 3D models level-of-detail. The leftmost model has the least number of polygons. White points on the model are data samples. The models are shaded according to surface normal profiles on the polygonal surfaces.

densely sampled set of states in parameter space are evaluated with these models; the bulk of very unlikely states are quickly discarded, while the remainder are subsequently be evaluated via the higher LOD models. By combining the 3D models with increasing LOD into a hierarchy, we form a detection cascade that can be globally optimized with respect to running time and overall detection performance. Our method does not rely on local search techniques and will not be trapped in local minima.

The consideration for using 3D models directly is motivated by the fact that a complete description of an object may not always be available. Existing works build a large database of 2D shape templates generated from a 3D model, with each template corresponds to a certain viewpoint of the object. In appearance-based methods, it is assumed that objects possess known surface properties that allow associations to some learned feature descriptors. However, object reflectance or emission information may not always be available or constant (for instance, in non-visible spectrum imagery, an object may have different appearance depending on its thermal profiles. Objects may also have very different appearance under varied lighting conditions). Current generative models [1, 2] do not handle significant lighting changes, and assume that features can be reliably detected by interest operators. One major limitation of generative models is that the learned classifiers cater only to single viewpoints (i.e. one set of model parameters for each different viewpoint of an object, even for mirror images of the object).

We solve the detection problem in an alignment framework. As in Viola's work [3], given vertices and their connections in a model, we derive surface normals and match distribution of the surface normals to the observed intensity using mutual information [4]. While Viola highlighted that their technique is purely intensity-based, we found that for reduced ambiguity, mutual information between projection contour of the 3D model and image edge maps can be included to help increasing detection performance. This is a crucial enhancement when mutual information is to be applied to real world scenes, out from the medical image registration domain where mutual information has enjoyed a great deal of success. We note that mutual information is chosen as the matching metric as it can be used for measuring similarity between multi-modal data, allowing the framework to be applied to multispectral imagery.

Matching polyhedral models of objects to images in order to recover pose parameters is a problem that has been tackled by many authors. The contributions of this work include:

- An alignment framework by maximization of mutual information with enhanced detection performance by including contour information.
- Speeding up the search in the 6D pose transformation space by using a cascade of 3D models of increasing levels-of-detail.

Section 2 reviews related work. We then discuss in section 3 about how non-uniform sampling is introduced to reduce the number of candidate hypotheses. In section 4, we present the cascaded detection strategy using an LOD hierarchy. Some experimental results and future work end the paper.

2 Previous Work

Campbell and Flynn provided a comprehensive survey of 3D object recognition techniques using 3D geometric models [5]. We discuss some previous alignment based work. Kollnig and Nagel [6] described a vehicle tracking system that fits discontinuities between surface facets of a simple polyhedral model to image gradients. A gradient image obtained from the discontinuities is matched to gray value gradients of the input image. The difference between the synthetic gradient image and the gray value gradient of the image is used to update the model pose. Tan *et al.* [7] described a vehicle detection system using simple polyhedral car models. Target objects are assumed to be lying on a known ground plane. This assumption reduces the problem of localization and recognition from 6 degrees-of-freedom to 3 degrees-of-freedom. The ground plane constraint allows pose to be estimated by matching 2D image and 3D model lines using Hough transform. Before line correspondences are established, the ground plane has to be recovered from the input image. Suveg and Gosselman [8] aligned simple polyhedral models to aerial views of buildings using mutual information as matching metric. Mutual information between gradient magnitude along model contour and image data is computed. If more images are available, mutual information between texture information of multiple images is included as additional information.

Our work is based on Viola's alignment approach [3]. Surface normals of the object are model instances and matched to intensity values by maximizing their mutual information with respect to a set of transformation parameters. Leventon *et al.* [9] extended the alignment framework to using multiple views of the object when single image does not provide enough information.

The notion of cascading has been applied to object detection [10]. In this work, the cascading of 3D LOD models for object detection in 2D images is a new idea, which aims to detect target objects and discard unlikely hypothesis rapidly.

3 Parameter Space Sampling

In this work, the state space comprises six parameters of 3D rigid-body transformation (three for translation and three for rotations). Existing work estimates pose parameters by optimization of an evaluation function. Such optimization-based methods have common problems of being sensitive to initial pose, and may experience slow convergence or be trapped in local minima. Viola [3] derived an approximation of the derivative of mutual information with respect to the transformation parameters, and used a stochastic gradient descent algorithm to seek the local maximum. Although stochastic gradient search is relatively fast as compared to techniques that do not require function derivatives (such as Powell's), it is still faced with the problem of local extrema.

In order to escape from the aforementioned problems, our framework falls back on exhaustive search. An exhaustive search in a discretized state space of the pose parameters will allow the global maximum to be obtained if the state space is sampled with sufficient resolution. Such exhaustive search does not have problems of being trapped at local maxima and do not depend on initial states, but can be enormously expensive for high-dimensional spaces.

3.1 Appearance-Dependent Sampling of State Space

Attempting to uniformly sample the full transformation state-space is inefficient as various different combinations of parameter values do not necessarily lead to significantly visible changes in the projected image space. In this section, we describe how the range and sampling intervals of parameters are manually determined in order to limit the sampling to pose variations of interest.

The range for each parameter is set depending on the visibility of the projections. For instance, the X-axis translation of the polyhedral model is placed in the range $\{-1.5, 1.5\}$ units with respect to a virtual camera of known focal length and viewing screen size. The object is either totally clipped or unrecognizable if the x parameter exceeds this range. The Y-axis translation has a range of between -1 and 1 while the range for Z-axis translation is $\{-2, 2\}$. The Y-axis rotation has the largest range as it involves greater variation in object appearance, i.e. the number of visible views corresponding to Y-axis rotation is larger than X- and Z-axis rotations. Both rotations about the X- and Z-axis have a range of between -10 and 10 degrees.

After determining the range for each parameter, we can further improve efficiency by setting different sampling scales for each parameter. One way of defining these step sizes is by looking at how different the models appear in image space when each of the parameter is changed: for instance, when the model is translated by one unit along the X-axis in the object space, how many pixels the model appears to have been translated in the X-axis direction in image space? Through such observation for all the parameters, we can define a step size ΔS_p for each parameter p, where each ΔS_p accounts for a cluster of parameter values with very similar appearance on the viewing screen.

In the next section, we describe detection using a cascade of increasing levels-of-detail of the object model.

4 Cascaded Detection Using a Level-of-Detail Model Hierarchy

We construct 3D models of different levels-of-detail (LOD) using a model simplification software [11], which reduces the number of polygons while maintaining high-quality approximation to the original polygonal surfaces. As models with lower LOD take much shorter time to render, these models are first used to evaluate the densely sampled states of parameter space in order to quickly discard the very unlikely states. However, as the accuracy of these lower LOD models are poorer, higher LOD models are required to further evaluate the more likely states. By combining the 3D models with increasing LOD into a hierarchy, we form a detection cascade.

Recent improvements in methods for the acquisition of 3D models allows for high-quality 3D models to be obtained more easily. Additionally, we use 3D models that are freely available from the Internet. Figure 1 illustrates an example LOD hierarchy with white dots on the models as locations where surface normals are sampled. Surface normals are collected from normal maps (images in figure 1 are normal maps) rendered using OpenGL, where (x, y, z) components of a normal correspond to (r, g, b) values of a point on the normal map. For a set of pose parameters P, the model has normal samples N and corresponding intensity values I. The mutual information MI between N and I is [4]:

$$MI(N, I) = H(N) + H(I) - H(N, I) \tag{1}$$

$H(A)$ is entropy for random variable A:

$$H(A) = -\sum_{a} p(a) \log p(a) \tag{2}$$

while $H(A, B)$ is joint entropy for random variables A and B that is defined as:

$$H(A, B) = -\sum_{a} \sum_{b} p(a, b) \log p(a, b) \tag{3}$$

As the lower LOD models are coarse shape approximations to the object, their MIs have lower values than MI for the model with the highest LOD. In addition, models of lower LOD are *weak* models as they may correspond to multiple objects (i.e. including non-target objects) in the image. In the initial levels, we use these weak models to discard unlikely states using a lower threshold value. State vectors that meet the threshold will get passed to the next level with a higher threshold value. As the weak models have lower rendering cost, detection in a cascade manner results in a speed up. Figure 2 shows the cascade architecture for a car model.

For a cascade $C = \{m_1, m_2, ..., m_n\}$, MIs between model m_i at level i and image are evaluated at the discrete 6D state vectors defined by the stratification. The point at which MI (computed using (1), in the same manner as Viola's algorithm [3]) is maximum is recorded, t_i. Given r training images, we run the same evaluation using model m_i for each training image and record its maximum MI values in T_{m_i}:

$$T_{m_i} = \{t_{m_{i1}}, t_{m_{i2}}, ..., t_{m_{ir}}\}$$

The average of T_{m_i} becomes the MI threshold value for level i.

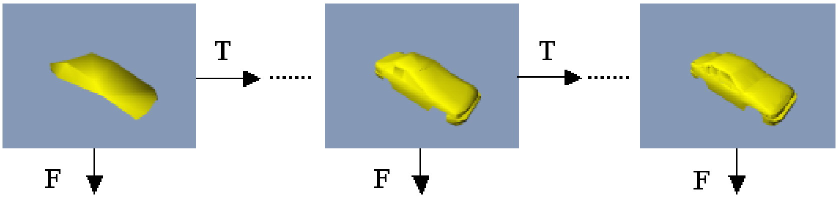


Fig. 2. A cascade of 3D models with increasing LOD, with each model acting as a classifier

5 Experiments

5.1 Normal Maps Generation

Surface normals are collected from visible surface patches for each hypothetical pose. While determining front-facing polygons is a simple task, it is non-trivial to determine visible polygons as occlusion has to be taken into account. We adopt the normal map generation method in computer graphics. Normals are collected from normal maps rendered using the methods described in [12]. Leventon *et al.* [9] also generated normal maps for MI computations. RGB channels of the normal maps correspond to (x, y, z) coordinates of surface normals (figure 1).

5.2 Edge Information for Reduced Ambiguity

Our evaluation function is the mutual information between object and image data as expressed in (1). Using intensity information alone in a single image may not be sufficient as shown by [9], as the observed data may not provide enough information due to occlusion, background clutters or variation in illumination condition. While Viola highlighted that their method is purely intensity-based, we found that to apply mutual information to real world scenes, we have to include other information so that the matching metric is more discriminative.

To illustrate the ambiguity issue, a model is rotated around the Y-axis and mutual information measures are recorded at uniform steps of five degrees from 0 to 180 degrees (figure 3). At one of the angles, the model is correctly aligned with the image. The graph shows that maximum mutual information does not occur at the ground truth (the shaded marker) but at a nearby pose (65).

To resolve this ambiguity, edge orientation for the projected contours of the model (figure 4) are added into the mutual information between model and image data. For each hypothetical pose, contours of the projected model are detected using an edge detector. Edge orientations of the model contours, EO_m, are computed. We then detect edges around the model contours on the *image*. Image edge pixels within a window (we used 10 pixels) around each edge pixel of the model contours are included in the calculation of the edge orientations for the image edges, EO_i. Mutual information between model and image edge orientations, $MI(EO_m, EO_i)$, is then added to $MI(N, I)$ defined in (1):

$$MI = MI(EO_m, EO_i) + MI(N, I) \tag{4}$$

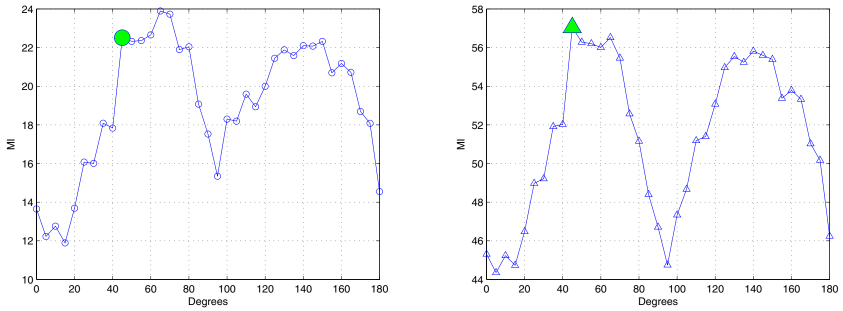


Fig. 3. (a) Intensity information alone is insufficient for matching using mutual information. (b)After adding in edge information into the objective function (1), maximum mutual information is achieved at the ground truth (i.e. the shaded marker).

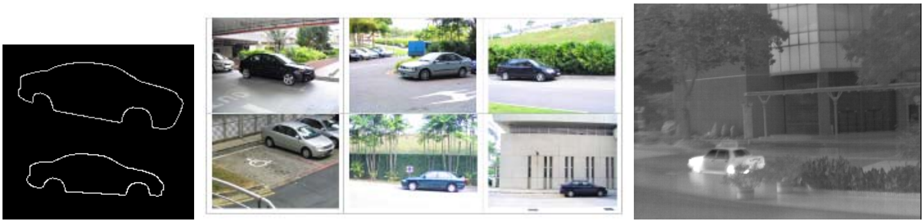


Fig. 4. (a) Edge orientation is included as additional information. (b) Some of the test images. (c) One of the infra-red images used in experiments.

Figure 3 shows that after adding in the edge information, the maximum mutual information occurs at the ground truth.

5.3 Error Analysis

To ensure that maximum mutual information (MI) appears at state vectors (which are defined by the stratification) near the ground truth, we examine the energy surface of the state vectors. Firstly, a plane d that cuts through the point with the maximum mutual information and the ground truth point is chosen. We then consider points near to plane d (points G) and project the points onto plane d. MI values versus 2D coordinates of points G projected on plane d is then plotted. A visualization of the energy surface is shown in figure 5. We found that with edge information added, the estimated pose is always at or close to the ground truth pose.

5.4 Object Detection

We applied our framework to vehicle detection. Software for constructing 3D models of different LOD are readily available on the Internet, such as the popular model simplification tool by Garland and Heckbert [11]. We used the MultiRes

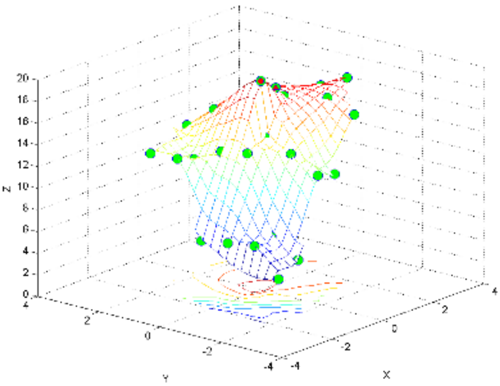


Fig. 5. Visualization of energy surface for pose parameters. Maximum mutual information (point with red diamond) appears near to the ground truth point (point with red circle).

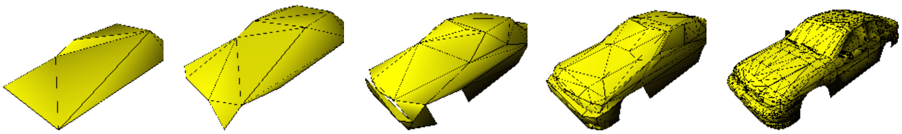


Fig. 6. 3D car models that form the LOD hierarchy in our experiments

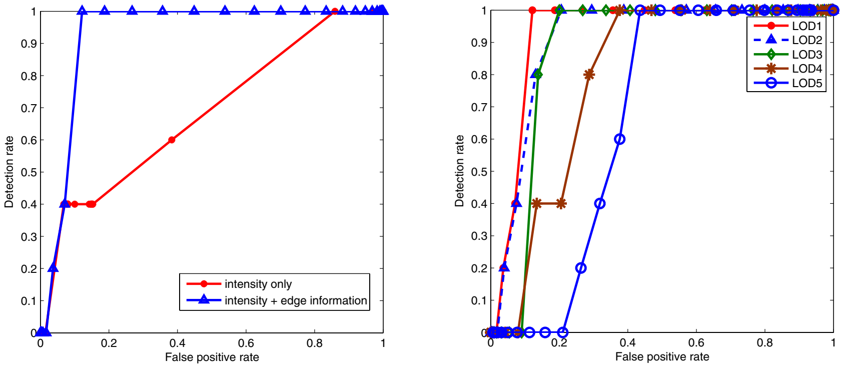


Fig. 7. (a) ROC curves for mutual information with and without edge information. (b) ROC curves for the LOD models, where LOD1 is the highest LOD model.

modifier in 3D Studio Max to generate the LOD models. The cascaded detection method was tested on both real and infra-red images (figure 4). To evaluate the performance of the detection algorithm, we first manually align the highest LOD model to the images and these ground truth poses (which are a few 6D state vectors) are then recorded as true positives. We then run through the cascade and record the number of hits and false positives. Figure 7 shows the receiver operating curves (ROC) for mutual information with and without edge information on one of the test images. The ROC curves show that by including edge information, detection performance improve significantly.

While an exhaustive search in the stratified parameter space using the highest LOD model (i.e. single layer) takes near to thirty minutes to complete, the cascaded detection takes about eight minutes using a hierarchy of five models. The car models have 13, 26, 78, 366, 3317 polygons respectively (figure 6). ROC curves for the five LOD models are shown in figure 7. We noticed that there is still room for improvement in speed, as currently the models are handled independently without considering their individual detection performance. This is a design issue of the cascade: choosing which model to be included in the hierarchy, and how to set the threshold value for each model by analyzing their ROC curves.

6 Conclusion

We have presented an alignment-based detection framework using a hierarchy of 3D models of increasing levels-of-detail. The designed cascade speeds up the search for the optimal pose parameters in a densely sampled parameter space. As the method does not face the issues of local optimum and convergence failures, it is more reliable and practical than methods that rely on directed search techniques. We have demonstrated that by adding edge information into the calculation of mutual information, discriminative power of the matching metric is increased significantly for real scenes.

We are working on an optimization framework for improving the design of the cascade such that optimal trade-off between performance and running time can be achieved. Choosing models at the optimal levels-of-detail to be included is part of the cascade design issue. We would also like to more extensively test the framework using other data set.

References

1. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Volume 2., Madison, WI (2003) 264–271
2. Weber, M., Welling, W., Perona, P.: Unsupervised learning of models for recognition. In: Proceedings of the European Conference on Computer Vision. Volume 1., Dublin, Ireland (2000) 18–32
3. Viola, P.: Alignment by Maximization of Mutual Information. PhD thesis, Massachusetts Institute of Technology (1995)
4. Cover, T., Thomas, J.: Elements of Information Theory. John Wiley (1991)
5. Campbell, R., Flynn, P.: A survey of free-form object representation and recognition techniques. Computer Vision and Image Understanding **81** (2001) 166–210

6. Kollnig, H., Nagel, N.N.: 3d pose estimation by directly matching polyhedral models to gray value gradients. International Journal of Computer Vision **23** (1997) 283–302
7. Tan, T., Sullivan, G., Baker, K.: Model-based localization and recognition of road vehicles. International Journal of Computer Vision **27** (1998) 5–25
8. Suveg, I., Gosselman, G.: Mutual information based evaluation of 3d building models. In: Proceedings of the International Conference on Pattern Recognition. Volume 3., Quebec City, Canada (2002) 188–197
9. Leventon, M., Wells III, W., Grimson, W.: Multiple view 2d-3d mutual information registration. In: DARPA IMage Understanding Workshop. (1997) 625–630
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 1., Kauai, HI (2001) 511
11. Garland, M., Heckbert, P.: Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. (1997) 209–216
12. Decaudin, P.: Cartoon-looking rendering of 3d scenes. Technical Report 2919, INRIA (1996)

Heuristic Pre-clustering Relevance Feedback in Region-Based Image Retrieval

Wan-Ting Su, Wen-Sheng Chu, and Jenn-Jier James Lien

Robotics Laboratory, Dept. of Computer Science and Information Engineering,
National Cheng Kung University,
No. 1, Ta-Hsueh Road, Tainan, Taiwan
{wangting, l2ior, jjlien}@csie.ncku.edu.tw
http://robotics.csie.ncku.edu.tw

Abstract. Relevance feedback (RF) and region-based image retrieval (RBIR) are two widely used methods to enhance the performance of content-based image retrieval (CBIR) systems. In this paper, these two methods are combined. And a region weighting scheme reflecting the process of human visual perception is also proposed to enhance the weighting importance assigned to the region whose pixels are closer to the attention center. Furthermore, rather than using a single positive feedback group, the proposed approach introduces RBIR to the relevance feedback with multiple positive and negative groups. To guide users in grouping the positive feedbacks, the proposed system provides a heuristic pre-clustering result automatically. Using these guiding clusters, the users can re-group the positive feedbacks to express his/her particular interests. Finally, Group Biased Discriminant Analysis (GBDA) is modified and applied to the similarity measure between images constructed on the basis of the region-based relevance feedbacks.

1 Introduction

Content-based image retrieval (CBIR) [1, 2, 3, 5, 6, 7, 9 10, 11, 12, 13, 16, 17, 18] is a technique used for extracting similar images from an image database. The most challenging aspect of CBIR involves the gap between high-level semantic concepts and low-level image features. In general, two approaches are commonly employed to reduce, or to bridge this gap.

The first approach involves the extraction of the region-based features to reflect the user's perception. Compared to the case where global image features [9, 12, 15] are considered, the region-based image retrieval system [1, 2, 3, 5, 6, 7, 11, 16, 17, 18] applies image segmentation to decompose an image into regions, which are closer to the perception of the human visual system. Some region-based image retrieval systems, like Netra [7], Blobworld [1], and IDQS [17], compare images based on individual region-to-region similarity. During retrieval, the user is provided with the segmented regions of an image and is required to assign several properties, such as the regions to matched, the matching regions, the number of expected regions, and even the feature weights of the regions. Such querying system provides the much power of control to the user. Nevertheless, automatic and semantically precise image segmentation is still an open problem, as discussed in [5]. For example, an image

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 294 – 304, 2006.

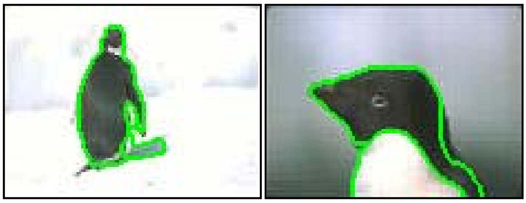


Fig. 1. Segmentation results of two images of a penguin

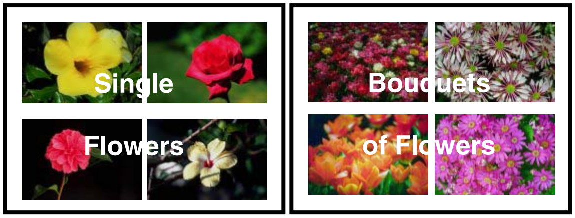


Fig. 2. Two groups: single flowers and bouquets of flowers

segmentation algorithm may decompose a penguin into one region (the entire penguin), but another penguin into two regions (the head and the body), as shown in Figure 1. Because of the difficulty of the accurate segmentation, it's not obvious for the user to choose a query region, especially for images without distinctive objects or scenes. To ensure robustness against inaccurate segmentation, the integrated region matching (IRM) algorithm [5] proposes an image-to-image similarity measure that combines the information from all of regions between images. Here, we adopt this approach to reduce the uncertainty of the region segmentation and to improve the retrieval performance.

The second approach taken to reduce the gap involves the use of relevance feedback. This approach employs an online learning scheme to improve the extraction performance by applying positive and negative examples according to the user's perception [3, 6, 10, 13]. Nakazato and Huang proposed a novel approach, Query-by-Groups [10], in which the user was provided with a mechanism to specify his/her interests in terms of multiple positive and negative image groups, as shown in Figure 2. In order to guide the user, a heuristic pre-clustering method is developed in this paper.

Although many relevance feedback methods using global features have developed, it's rarely applied to the RBIR system. In the retrieval system proposed in the current paper, we try to integrate the aforementioned two approaches, that is, the region-based image retrieval and relevance feedback with multiple positive and negative groups. The relevance feedback algorithm based on the GBDA approach is designed according to the characteristic of the region-based representation. Furthermore, a region weighting scheme, which mimics the process of human visual perception, is also proposed.

2 System Overview

Figure 3 presents a flow chart of our proposed system. During offline preliminary preparations, the features of the segmented regions and the region weights of all images in the database are extracted automatically. During online process, when a query image is supplied by the user, all of the images are sorted according to their similarity to this query. If the user is not satisfied with the extraction results, he or she can specify the feedbacks to refine the results in the next iteration. The system also provides heuristic pre-clusters to help the user group the positive feedbacks. The user can then manually revise the clusters based on the guiding cluster results.

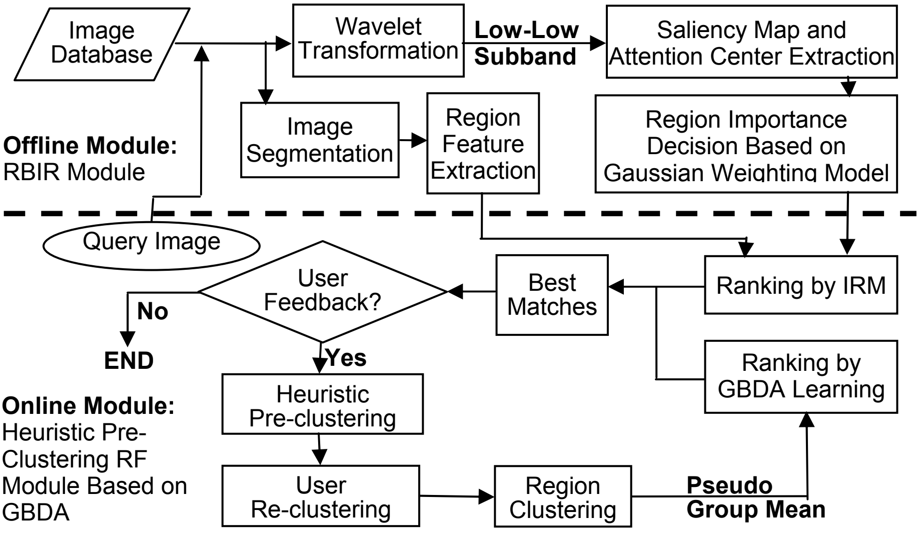


Fig. 3. Overview of the proposed system

3 Region-Based Image Retrieval

In a region-based image retrieval system, image segmentation is applied to decompose an image into several regions first. An image is represented by a set of regions, which contains its low-level features and importance weights. Then, both properties are used in the evaluation of the similarity between two images.

3.1 Image Segmentation

The segmentation algorithm we employed is based on local homogeneity analysis presented in [4]. The basic idea of defining the homogeneity of a pattern is to integrate the directional intensity changes of the surrounding pixels, which are located within a local window. Applying the criterion to the original image results in the H-image which is a gray-scale image whose pixel values are the Homogeneity values. The high and low values in the H-image point to the possible region boundaries and region interiors, respectively. Then, seeded regions with lower H values are chosen to do region growing method. Finally, the regions are merged based on their color histogram similarities to avoid over-segmentation and here an agglomerative algorithm is used. The segmentation result is shown in Figure. 1.

After locating the boundaries between segmented regions, we find that those pixels in the boundaries cannot be assigned unambiguously to regions. In order to make the description of regions more accurately, the boundary pixels are deleted when doing feature extraction. Deleting the ambiguous information of boundaries can lessen the uncertainty caused by image segmentation in the region-based image retrieval system.

3.2 Region Feature Extraction

In the current implementation, each region of images in the database is characterized by color and texture. The first two moments, mean and standard deviation, from each channel of HSV color space are extracted as color feature [15]. The texture feature is represented by the standard deviation of wavelet coefficients in 4 pyramids decorrelated subbands [14]. So the dimensionality of the visual feature spaces is 10.

3.3 Region Importance Decision

Here, two processes are considered. First, we try to extract the attention center of the entire image, which simulates the importance in the view of the human perception. Second, the Gaussian weighting model is proposed, which provides higher weights to the pixels near the attention center and lower weights to the pixels far from the attention center. Then, the region importance is decided by all pixel weights inside the region.

Attention Center Extraction. In [8], Ma et al. concluded that color contrast in an image plays the most important factor, which dominantly determines human visual perception. Therefore, they proposed an image attention analysis method involving the use of a contrast-based saliency map. In their approach, the contrast level in the saliency map was regarded as density and the attention center was represented by the centroid of saliency map.

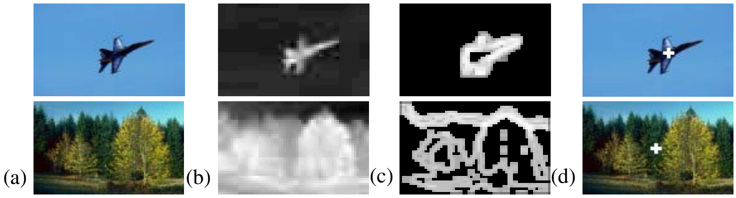


Fig. 4. Contrast-based image attention analysis: (a) original MxN-pixel (width*height) image, (b) M/2xN/2-pixel wavelet LL-subband, (c) M/2xN/2-pixel saliency map, and (d) extracted attention center of original image

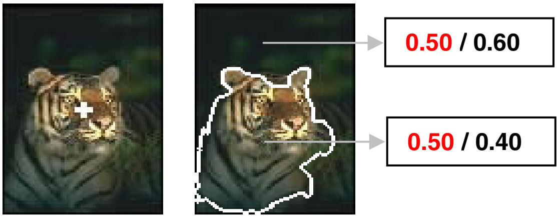


Fig. 5. Our region weight (left and red) and area percentage weight of an image

Wavelet transformation is widely applied in image processing since its properties of the multi-resolution decomposition can be adapted to describe image features. So to reduce the computational cost and to preserve the basic image content, contrast extraction is applied to the wavelet coefficient in the LL-subband, as shown in Figure 4(b). Subsequently, the image contrast is applied in the LUV color space.

The contrast value $\mathbf{C}_{i,j}$ of pixel **p** at image location (i, j) is defined as [8]:

$$C_{i,j} = \sum_{q \in \Theta} d(p_{i,j}, q) \tag{1}$$

where the intensity difference **d** is computed by Gaussian distance, **θ** is the neighborhood area and **q** is the neighborhood pixel of pixel (i, j). From pixel-to-pixel contrast addition, $C_{i,j}=C_{i,j}(L) + C_{i,j}(U) + C_{i,j}(V)$. Furthermore, normalizing the contrast values for all of the pixels to the scale [0, 255] generates a saliency map, as shown in Figure 4(c).

From [8], the attention center $(\boldsymbol{x_0}, \boldsymbol{y_0})$ can be computed by equation (2), i.e.

$$\begin{cases} x_0 = \dfrac{1}{C_M} \displaystyle\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} C_{i,j} \times i \\ y_0 = \dfrac{1}{C_M} \displaystyle\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} C_{i,j} \times j \end{cases} \tag{2}$$

where $C_M = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} C_{i,j}$ is the 0th order moment of the saliency map and the image size is ***MxN***. Figure 4(d) provides two illustrative examples of extracted attention centers.

Gaussian Weighting Model. Our basic assumption is that the pixel closer to the attention center has higher importance. We consider the Gaussian model of the distance between the pixel and the attention center to evaluate the pixel importance $\boldsymbol{PI_i}$ defined by the following equation.

$$PI_i = \exp(-dis(i, C_0) / \sigma) \tag{3}$$

where ***dis*** is the Euclidean distance of the location difference between pixel ***i*** and attention center $\boldsymbol{C_0}$,and $\boldsymbol{\sigma}$ is the standard deviation of all distances between each pixel in the entire image and attention center. Then by considering the region sizes, the region importance $\boldsymbol{w_i}$ is defined as the summation of the importance of pixels inside the region ***R***, i.e.

$$w_i = \sum_{j \in R_i} PI_j \tag{4}$$

Figure 5 shows an example of region weights using our weighting scheme (left and red) and using area percentage (AP). In the human visual perception, the weight of the tiger in the image should take higher importance. Obviously, the weight of the tiger using our weighting scheme is higher than AP by 10%.

3.4 Image Similarity Measure

Considering that the image segmentation may not be perfect, as shown in Figure 1. The integrated region matching (IRM) [5] allows one region of an image to be matched to several regions of another image. Compared with image retrievals based on individual region-to-region similarity, IRM is robust to inaccurate image segmentation.

Assume that image $\boldsymbol{I_P}$ contains $\boldsymbol{m}$ regions and image $\boldsymbol{I_Q}$ contains $\boldsymbol{n}$ regions. A matching between regions $\boldsymbol{p_i}$ and $\boldsymbol{q_j}$ is assigned a significance credit $\boldsymbol{s_{i,j}}$, where this credit represents the importance of the matching in determining the similarity between the two images. Furthermore, let d($\boldsymbol{p_i}$, $\boldsymbol{q_j}$), the region feature distance between $\boldsymbol{p_i}$ and $\boldsymbol{q_j}$, be the Euclidean distance. The IRM distance between $\boldsymbol{I_P}$ and $\boldsymbol{I_Q}$ is given by the weighted sum of all the similarities between the region pairs, i.e.

$$d(I_P, I_Q) = \sum_{i=1}^{m}\sum_{j=1}^{n} s_{i,j} d(p_i, q_j) \tag{5}$$

Therefore, the problem of defining the similarity between the two images then becomes one of choosing the significance credit of all of the region pairs. More details on the IRM can be found in [5].

4 Heuristic Pre-clustering Relevance Feedback Based on GDBA

In interactive region-based or content-based image retrieval processes, the system must re-calculate the similarity and the feature weight based on the user's feedbacks [10]. This study modifies the GBDA method proposed in [10] to re-estimate both the similarity and the feature weight. In other words, the image similarity ranking process can be reduced to an online calculation of the feature space discriminating transformation matrix.

4.1 Guiding Pre-clustering

For a typical user, grouping all of the positive images is not an intuitive process. Therefore, the system proposed in this paper provides heuristic pre-clusters to assist him or her in manually grouping these positive feedbacks.

The proposed on-line guiding pre-clustering algorithm commences by computing and sorting the IRM distances between any two positive feedbacks. The two images with the minimal IRM distance are grouped as the first positive class. Subsequently, the image with the shortest distance from one of the positive examples in the first class is chosen. If when adding this positive example to the current positive class, the sum of all of the distortion between the images in the positive class is less than a predefined threshold, then this image can be inserted into the current positive class; else a new class is created for this positive example. These processing steps are repeated iteratively until each of the positive feedbacks has been assigned to a corresponding class.

4.2 Region-Based Relevance Feedback Based on Group Biased Discriminant Analysis (GDBA)

Briefly, GBDA attempts to cluster each positive class (or group), while scattering negative examples (or samples) away from each positive class. GBDA achieves this via the following equation,

$$\overline{W} = \arg\max_{W} \left| \frac{W^T S_{PN} W}{W^T S_W W} \right| \tag{6}$$

where $\boldsymbol{S_w}$ is the sum of the within-class scatter matrix of the positive groups and $\boldsymbol{S_{PN}}$ is the sum of the between-class scatter matrix of the positive-to-negative groups.

Pseudo Group Mean Representation. Assume that all regions of the examples in the positive class can represent this class. We simply combine all of the regions in the positive class into one region set, which is regarded as the pseudo mean of the positive class. In order to fit the constraint of the region importance, the total importance of the pseudo mean should be normalized to 1. Suppose there are $\boldsymbol{n}$ positive examples in one of the positive classes. Hence, the summation of total region importance of the pseudo mean is $\boldsymbol{n}$. To satisfy the constraint, we simply set the region importance of the pseudo mean $\boldsymbol{w_m}$ as follows:

$$w_{m\,k}^{\;i} = w_k^i / n \tag{7}$$

where w_k^i is the i^{th} region importance of the k^{th} example in the positive class.

Region Clustering. As the number of the feedback iteration increases, the number of regions of the positive examples increases rapidly. Furthermore, the executing time to compare similarity between images is proportional to the number of regions in the images. Consequently, to avoid the retrieval speed slowing down, the regions with similar low-level feature vectors are merged together via clustering. Here, the k-means algorithm is adopted to group the regions of the examples in the same positive class into a few clusters, each of which represents a new region of the pseudo mean. We adaptively choose the number of clusters $\boldsymbol{k}$ by gradually increasing its value. $\boldsymbol{k}$ is initialized to 2 and increases by 1 at each step. The process stops if the average distortion between all the positive regions and their nearest cluster centers is below a threshold, which can be adjusted according to the experiments. Here, the threshold is set to 0.01. After clustering, the average feature of the regions in the same cluster is viewed as the feature of the new region. The new region importance is the summation of all region importance in the same cluster.

Region-Based GBDA Formulation. Finally, we define the terms in (6) as follows,

$$S_w = \sum_{i=1}^{c} S_{Pi} \tag{8}$$

$$S_{Pi} = \sum\nolimits_{x_p \in C_k} S_{i,j} (x_p^i - q_k^j)(x_p^i - q_k^j)^T \tag{9}$$

$$S_{PN} = \sum_{k=1}^{c} S_{Nk} \tag{10}$$

$$S_{Nk} = \sum\nolimits_{y_n \in D} S_{i,j} (y_n^i - q_k^j)(y_n^i - q_k^j)^T \tag{11}$$

where x_p^i is the i^{th} region of the p^{th} examples of the positive group $\boldsymbol{C_k}$, y_n^i is the i^{th} region of the n^{th} examples of the negative class, q_k^j is the j^{th} region of pseudo mean of the k^{th} positive class $\boldsymbol{C_k}$, $\boldsymbol{S_{i,j}}$ is the significance between the i^{th} region of an example and the j^{th} region of the pseudo mean, $\boldsymbol{c}$ is the number of positive groups and $\boldsymbol{D}$ is the set of negative examples. Here, we consider a negative example as one negative class.

As in Fisher's Discriminant Analysis (FDA), $\overline{W}$ is solved as the generalized eigenvector, $\boldsymbol{w_i}$, associated with the largest eigenvalue, $\boldsymbol{\lambda_i}$, i.e.

$$\lambda_i S_w w_i = S_{PN} w_i \tag{12}$$

If $\boldsymbol{S_w}^{-1}$ exists, a solution for $\overline{W}$ can be found by solving $\lambda_i w_i = (S_w^{-1} S_{PN}) w_i$. Therefore, the discriminating transformation matrix $\boldsymbol{A}$ becomes,

$$A = \Phi \Lambda^{1/2} \tag{13}$$

where $\boldsymbol{\Phi}$ is the matrix whose columns are the eigenvectors of $(S_w^{-1} S_{PN})$ and $\boldsymbol{\Lambda}$ is the diagonal matrix of the corresponding eigenvalues. Once the transformation matrix is available, the distance of the similarity measurement between two images (or samples), $\boldsymbol{x}$ with m regions and $\boldsymbol{y}$ with n regions, can be defined as:

$$distance(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} S_{i,j} (x_i - y_j)^T A (x_i - y_j) \tag{14}$$

Using this expression, the distance between images in the database and the pseudo mean of each positive group can be compared and sorted.

Table 1. Image Categories

1. Sunset	2. Flower	3. Car	4. Ape	5. Mountain
6. Penguin	7. Tiger	8. Bird	9. Horse	10. Building

Table 2. Average precision comparison between Gaussian weight (G) model and area percentage (AP)

	Top10	Top20	Top30	Top40	Top50	Top60	Top70	Top80	Top90	Top100
G	0.583	0.498	0.447	0.411	0.384	0.361	0.341	0.325	0.310	0.297
AP	0.579	0.491	0.441	0.404	0.377	0.354	0.334	0.318	0.303	0.291

5 Experimental Results

To evaluate the retrieval performance of the proposed system, this study considered a COREL image database containing 17695 images. In the experiments, a subset consisting of 1000 images from 10 selected categories is considered. Each category contains 100 images, each of which can be the candidate of the query image. A retrieved image is considered as relevant if it belongs to the same category of the query image. For each individual category, the retrieval accuracy is computed as the average precision rate of the top $\boldsymbol{N}$ retrieved images by retrieving 100 times. The selected categories are listed in Table 1.

5.1 Region-Based Image Retrieval Evaluation

Here, the region-based image retrieval is compared with the typical global representation [9, 12, 15]. To be fair, the same features are used for both representations. That

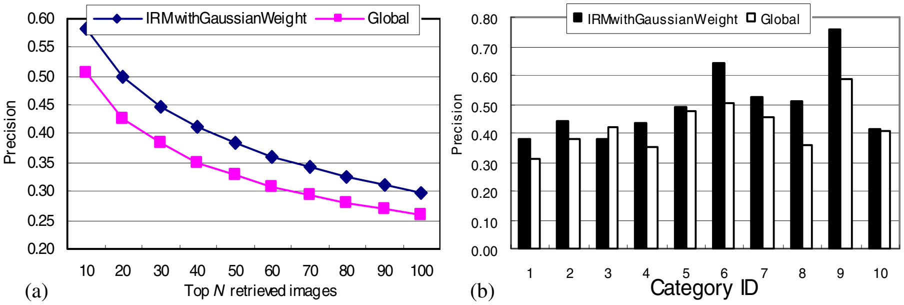


Fig. 6. (a) Comparison between global representation and region representation (b) Comparison between global representation and region representation by retrieving top 20 images in each category

is, the Euclidean distance for the similarity measure based on color and texture features described in section 3.2 is adopted. The result of both evaluations is shown in Figure 6(a). Obviously, the average retrieval accuracy of total 1000 queries of region-based retrieval is superior to that of global representation. Figure 6(b) shows the average accuracy of each category in the case of retrieving top 20 images. For the categories of *penguin*, *bird* and *horse*, which contain simpler backgrounds, the accuracy of region-based retrieval is higher than that of global representation by approximately 15%. But there are still some categories, which are not suitable for region-based retrieval, like the category of *car*. The reason may be that it contains complicated scenes (backgrounds) and diverse features under different poses.

5.2 Gaussian Weighting Model Evaluation

As we can see from Figure 5, the importance of the tiger region using the proposed weighting scheme is higher than that assigned in the area percentage (AP) method. To demonstrate the influence of Gaussian weighting model, Table 2 illustrates the performance of the initial retrieval result using the Gaussian weighting model (G) and the AP method. The average precision of total 1000 queries in 10 categories shows that the proposed weighting scheme is slightly better than that of the AP method. The reason might be because the Gaussian weighting model is dominated by the region size in current 1000 images, so its performance improvement is limited.

5.3 Gaussian Weighting Model Evaluation

To evaluate the integrated region-based image retrieval and GBDA, the performance of GBDA using global representation was compared. The same color and texture features described in section 3.2, were adopted. Gaussian weighting scheme was chosen as the weight of regions. To simulate the user's feedback, all of the retrieved results in the same category as that of the initial query image were regarded as positive examples, while those images in categories different from that of the query were regarded as negative examples. The other larger query set consisting of 7000 images under 50 categories from the COREL database is also considered and denoted as QS2.

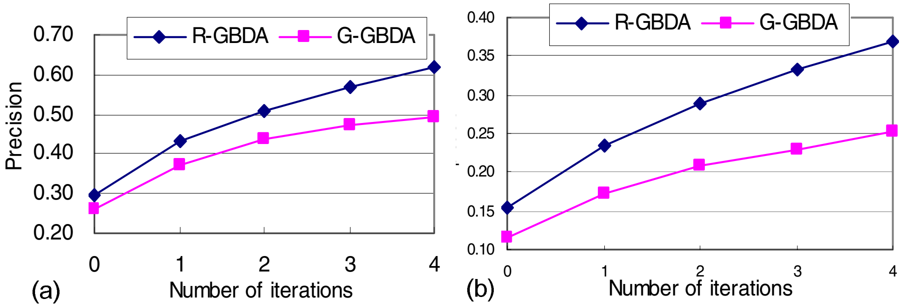


Fig. 7. (a) Accuracy comparison of two relevance feedback algorithms: R-GBDA and G-GBD A denote the GBDA algorithm using region-based retrieval and global-based retrieval, respectively. (b) Accuracy comparison of two relevance feedback algorithms on QS2: R-GBDA and G-GBDA denote the GBDA algorithm using region-based retrieval and global-based retrieval, respectively.

The results for both query sets are shown in Figures 7(a) and Figure 7(b). As shown in the figures, the performance of GBDA using region-based retrieval is better than that of GBDA using global representation by 12.42% (QS2: 11.71%) after four feedback iterations.

6 Conclusion

The major contribution in this study is integrating RBIR with the relevance feedback algorithm using multiple positive and negative groups. Compared to a single region matching scheme, the overall similarity measure can lessen the user's burden and reduce the uncertainty of the automatic region segmentation. A region weighting scheme based on human visual perception is introduced by utilizing the property of the color contrast saliency map. In addition, color contrast extraction is conducted in the wavelet LL-subband, which not only preserves the basic content of the image but also lowers the computational cost significantly.

The proposed system guides the user in clustering the positive feedbacks by providing heuristic pre-clustering results. The user can then revise the clusters manually by referring to the guiding cluster results. In order to obtain the scatter degree of the positive groups, all the regions of the positive examples in the group are combined into a region set representing the pseudo mean of that group. The k-means algorithm is adopted to accelerate the feedback process. Finally, the similarity between the query and the other images in the database is obtained by region-based Group Biased Discriminant Analysis.

In the future study, the authors intend to refine the retrieval performance of the developed system by assigning the importance of regions on the basis of user feedback information. Other features of the images, such as the region shape information or the spatial (or geometric) relationship between regions, and the use of keywords will also be considered.

Acknowledgement

The authors would like to thank Q. Tian, M. Nakazato and T.S. Huang for their source codes of GBDA.

References

1. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," in *IEEE Trans. on PAMI*, vol. 24, No.8, pp. 1026-1038, 2002.
2. Y. Chen and J. Z. Wang, "A Region-Based Fuzzy Feature Matching Approach to Content-Based Image Retrieval," in *IEEE Trans. on PAMI*, vol. 24, No.9, pp. 1252-1267, 2002.
3. F. Jing, M. Li, H.J. Zhang, and B. Zhang, "Relevance Feedback in Region-Based Image Retrieval," in *IEEE Trans. on CSVT.*, vol. 14, no. 5 , pp. 672-681, May 2004.
4. F. Jing, M. Li, H.J. Zhang, and B. Zhang, "Unsupervised Image Segmentation Using Local Homogeneity Analysis," in *Proc. IEEE Int. Symp. on Circuits and Systems*, vol. 4, pp. 145–148, 2002.
5. J. Li, J.Z. Wang, and G. Wiederhold, "IRM: Integrated Region Matching for Image Retrieval," in *Proc. of the 8th ACM Int. Conf. on Multimedia*, pp. 147-156, Oct. 2000.
6. V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Region-based Image Retrieval Using an Object Ontology and Relevance Feedback," in *Eurasip Journal on Applied Signal Processing*, vol. 2004, No. 6, pp. 886-901, 2004.
7. W.Y. Ma and B.S. Manjunath, "NETRA: A Toolbox for Navigating Large Image Databases," in *Proc. IEEE Int. Conf. on Image Processing*, vol. I, Santa Barbara, CA, pp. 568–571, Oct. 1997.
8. Y.F. Ma, and H.J. Zhang, "Contrast-Based Image Attention Analysis by Using Fuzzy Growing," in *Proc. of the 11th ACM Int. Conf. on Multimedia*, pp. 734-381, Nov. 2003.
9. W. Niblack *et al.*, "The QBIC Project: Querying Images by Content Using Color, Texture, and Shape," in *Proc. SPIE*, vol. 1908, San Jose, CA, pp. 173–187, Feb. 1993.
10. M. Nakazato, and T.S. Huang, "Extending Image Retrieval With Group-Oriented Interface," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, vol. 2, pp. 201-204, Aug. 2002.
11. A. Natsev, R. Rastogi, and K. Shim, "WALRUS: A Similarity Retrieval Algorithm for Image Databases," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, pp. 395–406, 1999.
12. A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based Manipulation of Image Databases," in *Proc. SPIE Storage and Retrieval for Image and Video Databases II*, San Jose, CA, pp. 34–47, Feb. 1994.
13. Y. Rui, T.S. Huang, M. Ortega, and Sharad Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," in *IEEE Trans. on CSVT*, vol. 8, pp. 644-655, Sep. 1998.
14. J.R. Smith, and S.F. Chang, "Transform Features for Texture Classification and Discrimination in Large Image Databases," in *Proc. IEEE Int. Conf. on Image Proc.*, pp. 407-411, Nov. 1994.
15. M. Stricker, and M. Orengo, "Similarity of Color Images," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, pp. 381-392, Feb. 1995.
16. Y. Sun, and S. Ozawa, "Semantic-meaningful Content-Based Image Retrieval in Wavelet Domain," in *Proc. of the 5th ACM SIGMM Int. Workshop on Multimedia Info. Retrieval*, pp. 122-129, Nov. 2003.
17. M. E. Wood, N. W. Campbell, and B. T. Thomas, "Iterative Refinement by Relevance Feedback in Content-based Digital Image Retrieval," in *Proc. of the 5th ACM Int. Conf. on Multimedia*, Bristol, U.K., pp. 13–20, Sept. 1998.
18. H.W. Yoo, S.H. Jung, D.S. Jang, and Y.K. Na, "Extraction of Major Object Features Using VQ Clustering for Content-based Image Retrieval," in *PR*, pp. 1115-1126, 2002.

Biologically Motivated Perceptual Feature: Generalized Robust Invariant Feature

Sungho Kim and In So Kweon

Dept. of EECS, Korea Advanced Institute of Science and Technology,
373-1 Gusong-Dong, Yuseong-Gu,
Daejeon, Korea
{sunghokim, iskweon}@kaist.ac.kr

Abstract. In this paper, we present a new, biologically inspired perceptual feature to solve the selectivity and invariance issue in object recognition. Based on the recent findings in neuronal and cognitive mechanisms in human visual systems, we develop a computationally efficient model. An effective form of a visual part detector combines a radial symmetry detector with a corner-like structure detector. A general context descriptor encodes edge orientation, edge density, and hue information using a localized receptive field histogram. We compare the proposed perceptual feature (G-RIF: generalized robust invariant feature) with the state-of-the-art feature, SIFT, for feature-based object recognition. The experimental results validate the robustness of the proposed perceptual feature in object recognition.

1 Introduction

A successful object recognition system should have proper balance between selectivity and invariance. Selectivity means the system has to discriminate between different objects or parts. Invariance means that the same objects or parts have to be invariant to photometric and geometric variations. It is generally accepted that the local invariant feature-based approach is very successful in this aspect. This approach is generally composed of visual part detection, description, and classification. The first step of the local feature-based approach is visual part detection. Schmid et al. [1] compared various interest point detectors and concluded that the scale-reflected Harris corner detector is most robust with respect to image variations. Mikolajczyk and Schmid [2] also compared visual part extractors and found that a Harris-Laplacian based part detector is suitable for most applications. Recently, several visual descriptors have been proposed [3][4][5][6][7] to encode local visual information of spatial orientation or edge density.

The key idea is the practical adaptation of tune-max property in receptive field to solve both the selectivity and invariance. We propose a similar computationally efficient model and finally compare it with the state-of-the-art method in computer vision technology.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 305–314, 2006.

2 Mechanisms of Receptive Field

2.1 Simple Cells/Complex Cells in V1

Simple cells and complex cells exist in the primary visual cortex (V1), which detects low level visual features. It is well known that the response of simple cells in V1 can be modeled by a set of Gabor filters eq. 1 [8]:

$$G(x, y, \theta, \varphi) = e^{-\frac{(x'^2+\gamma y'^2)}{2\sigma^2}} cos(2\pi\frac{x'}{\lambda} + \varphi) \qquad (1)$$

where $x' = xcos\theta + ysin\theta$ and $y' = -xsin\theta + ycos\theta$. According to recent neurophysiological findings [9], the range of an effective spatial phase parameter is $0 \leq \varphi \leq \pi/2$ due to symmetry. An important finding is that the distribution of spatial phase is bimodal, with cells clustering near 0 phase (even symmetry) and $\pi/2$ phase (odd symmetry).

Complex cell response at any point and orientation combines the simple cell responses. There are two kinds of models: weighted linear summation and MAX operation [8]. However, the MAX operation model has the most support since neurons performing a MAX operation have been found in the visual cortex [10]. The role of the simple cell can be regarded as a tuning process, that is, extracting all responses by changing Gabor parameters. The role of a complex cell can be regarded as a MAX operation from the tuning responses by selecting maximal responses. The former gives selectivity to the object structure and the latter gives invariance to geometric variations such as location and scale changes.

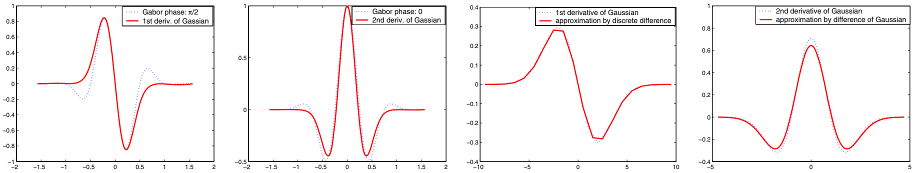
2.2 Receptive Field in V4

Through the simple and the complex cells, orientation response maps are generated and fine orientation adaptation occurs on the receptive field within the attended convex part in V4 [11]. The computational method of orientation adaptation phenomenon is steering filtering [12]. Adapted orientation is calculated by the maximum response spanned by basis responses ($tan^{-1}(I_y/I_x)$). There are also color blobs in a hyper column where the opponent color information is stored. Hue is invariant to affine illumination changes and highlights. Orientation information and color information is combined in V4 [13].

How does the human visual system (HVS) encode the receptive field responses within the attended convex part? Few facts are known on this point, but it is certain that larger receptive fields are used, representing a broader orientation band [14].

3 Visual Part Detection

What is a good object representation scheme comprising both selectivity and invariance? The global description with the perfect segmentation may show very



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

Fig. 1. Gabor filters are approximate by derivatives of Gaussian: $pi/2$ phase Gabor by 1st deriv. of Gaussian (a) and 0 phase Gabor by 2nd deriv. of Gaussian (b). The 1st and 2nd deriv. of Gaussian are further approximated by discrete difference of Gaussian (c) and by difference of two Gaussian kernels, respectively for computational efficiency.

good selectivity. However, this representation does poorly with respect to invariance since a perfect segmentation is impossible, and further, is sensitive to visual variation in light, view angle, and occlusion. Objects representation as the sum of sub-windows or visual parts may be a plausible solution and supported by the recognition by component (RBC) theory [15]. The main issues of this approach are how to select the location, shape, and size of a sub-window and what information to be encoded for both selectivity and invariance.

In this section, we present a visual part detection method by applying the tune-MAX [8] to the approximated Gabor filter. Serre and Riesenhuber modeled the Gabor filter using a lot of filter banks while changing scale and orientation. Furthermore, they fixed the phase to 0 which is another limitation. As described in Sect. 2.1, the distribution of spatial phase in receptive field is bimodal, 0 (even symmetry) and $\pi/2$ (odd symmetry). So, we can approximate the Gabor function by two bases generated by the Gaussian derivatives shown in Fig. 1(a) 1(b). The 1st and 2nd derivatives of Gaussian which approximate odd and even symmetry, respond to edge structures and convex (or bar) structures respectively.

The location and size invariance is acquired by the MAX operation of various tuning responses from the 1st, 2nd derivatives of Gaussian. Fig. 2 shows the complex cell responses using the approximated filters. The arrows represent the tuning process and the dot or circle represents the MAX operation.

(1) Location tuning using the 1st derivative of Gaussian:

- Select maximal response in all orientations within a 3×3 complex cell (pixel).
- Suitable method: Harris corner or KLT corner extraction (both eigenvalues are large).

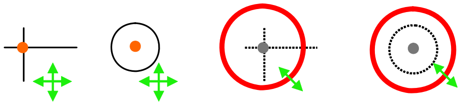


Fig. 2. (Left two) Interest points are localized spatially by tune (green)-Max (red) of 1st and 2nd derivative of Gaussian respectively. (Right two) Region sizes around interest points are selected by tune-Max of convexity in scale-space.

(2) Location tuning using the 2nd derivative of Gaussian:

- Select maximal response in all orientations within 3×3 complex cell.
- Suitable method: Laplacian or DoG gravity center (radial symmetry point).

(3) Scale tuning using the convexity:

- Select maximal response in directional scale-space [16].
- For computational efficiency, a convexity measure such as DoG is suitable. This is related to the properties of V4 receptive field where convex part is used to represent visual information [17].

For the efficient computation of the tune-Max, we utilize three approximation schemes: the scale-space based image pyramid [3], discrete approximation of the 1st derivative of Gaussian by subtracting neighboring pixels in a Gaussian smoothed image, and the approximation of the 2nd derivative of Gaussian (Laplacian) by difference of Gaussian (DoG). As shown in Fig. 1(c) 1(d), the kernel approximations for the Gabor bases are almost identical to the true kernel function. Fig. 3 shows the structure of our part detection method which computes the 1st and 2nd derivatives of Gaussian by subtracting neighboring pixels in a scale-space image and by subtracting between scale-space images, respectively. We calculate local max during scale selection. Fig. 4 shows a sample result of the proposed perceptual part detector. Note that the proposed method extracts complementary visual parts. We can get corner-like parts through the left path, and radial symmetry parts through the right path in Fig. 3 (See eq. 2). This is supported by the psychophysical fact that HVS attends to gravity centers and high curvature points [18] and objects are deconstructed into perceptual parts that are convex [19][20].

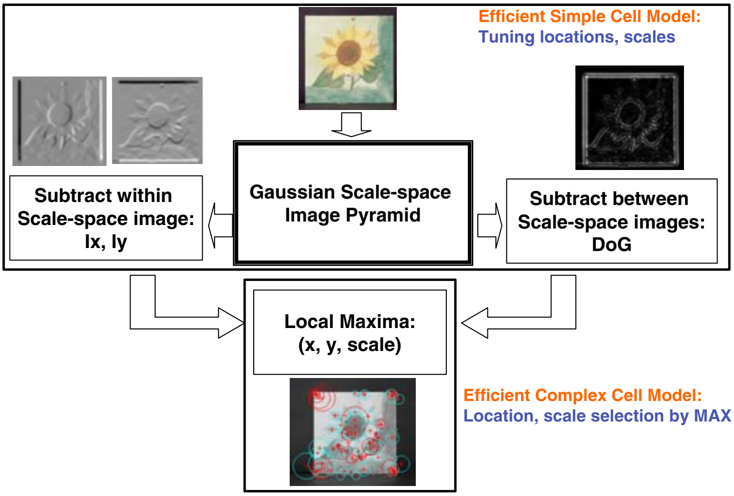

Efficient Simple Cell Model:
Tuning locations, scales
Subtract within Scale-space image: Ix, Iy
Gaussian Scale-space Image Pyramid
Subtract between Scale-space images: DoG
Local Maxima: (x, y, scale)
Efficient Complex Cell Model:
Location, scale selection by MAX

Fig. 3. Computationally efficient perceptual part detection scheme

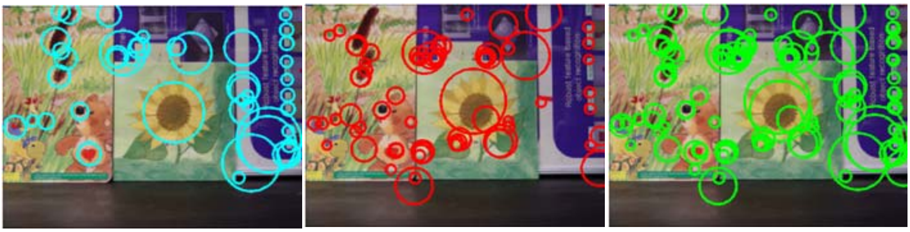


Fig. 4. The proposed part detector can extract radial symmetry parts (left), corner-like part (middle), and both of them (right)

$$\mathbf{x} = \max_{\mathbf{x}\in W}\{DoG(\mathbf{x},\sigma) or HM(\mathbf{x},\sigma)\}, \sigma = \max_{\sigma}\{DoG(\mathbf{x},\sigma)\} \tag{2}$$

where $DoG(\mathbf{x},\sigma) = |I(\mathbf{x}) * G(\sigma_{n-1}) - I(\mathbf{x}) * G(\sigma_n)|$ and $HM(\mathbf{x},\sigma) = det(\mathbf{C}) - \alpha trace^2(\mathbf{C})$. $\mathbf{C}$ is defined as eq. 3.

$$\mathbf{C}(\mathbf{x},\sigma) = \sigma^2 \cdot G(\mathbf{x},3\sigma)) \cdot \begin{bmatrix} {I_x}^2(\mathbf{x},\sigma) & I_x I_y(\mathbf{x},\sigma) \\ I_x I_y(\mathbf{x},\sigma) & {I_y}^2(\mathbf{x},\sigma) \end{bmatrix} \tag{3}$$

where $I_x(\mathbf{x},\sigma) = \{S([x+1,y],\sigma) - S([x-1,y],\sigma)\}/2$, $I_y(\mathbf{x},\sigma) = \{S([x,y+1],\sigma) - S([x,y-1],\sigma)\}/2$, $S(\mathbf{x},\sigma) = I(\mathbf{x} * G(\sigma)$.

4 Perceptual Part Descriptor

As we discussed in Sec. 2.2, we can mimic the role of receptive field V4 to represent visual parts. In V4, edge density map, orientation field, and hue field coexist in the attended convex part. These independent feature maps are detected from V1 (in particular, edge orientation and edge density is extracted using the approximated Gabor with $\pi/2$ phase).

Now the question becomes: How to encode the independent feature maps? We utilize the fact that larger receptive fields are used with a broader orientation band [14] and independent feature maps are combined to make more informative features [13]. Fig. 5(a) shows several possible patterns of receptive field in V4. The density of the black circle depicts the level of attention of the HVS in which 86% of fixation occurs around the center receptive field [18]. Each black circle stores the visual distributions of edge density, orientation field, and hue field of pixels around the circle. Fig. 5(b) shows how it works. Each receptive field stores them in the form of a histogram which shows good balance between selectivity and invariance by controlling the bin size, and is partially supported by the computational model of multidimensional receptive field histograms [21]. We can control the resolution of each receptive field such as the number of edge orientation bins, hue orientation bins except edge density which is scalar. Each localized sensor gathers information about edge density, edge orientation, hue color of receptive field. The histogram of an individual sensor is generated by simply counting the corresponding bins according to feature values weighted by

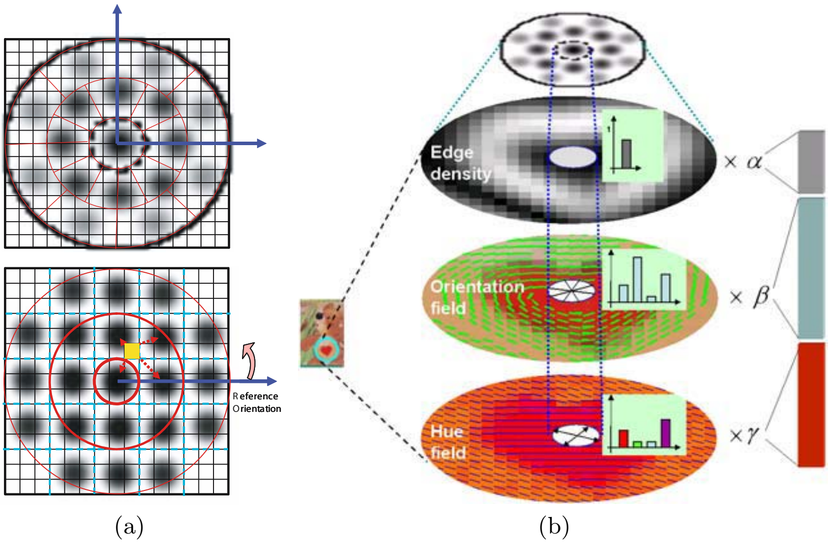


Fig. 5. (a) Plausible receptive field model in V4: (top) attention-based receptive fields (bottom) computationally easy receptive field pattern (b) Three kinds of localized histograms are integrated to describe local region

the attention strength and sensitivity of sensor. Scalar edge density is generated from edge magnitudes. This process is linear to the number of pixels within a convex part. Each pixel in a receptive field affects to neighboring visual sensors. After all the receptive field histograms are generated, we normalize each histogram vector. After we multiply these histograms with component weights ($\alpha+\beta+\gamma=1$), we integrate three kinds of features as Fig. 5(b) right column. Finally, we renormalize the feature (dim.: 21*(4+1+4)=189) so that the feature's energy equals to 1.

It is very important to align the receptive field patterns to the dominant orientation of an attended part if there is image rotation. We compared four kinds of dominant orientation detection methods: Eigenvector, weight steerable filter [12], maximum of orientation histogram [3], and radon transform. We found that the weighted steerable filter method showed the best matching rate in rotated images.

5 Experimental Results

We dubbed the proposed perceptual feature (perceptual part detector with generalized descriptor of pixel information) as the Generalized Robust Invariant Feature (G-RIF). In this section, we evaluate G-RIF in terms of object recognition. We adopt a new feature comparison measure in terms of object labeling. We use the accuracy of detection rate which is widely used in classification or labeling problems. Although there are several suitable open object databases such

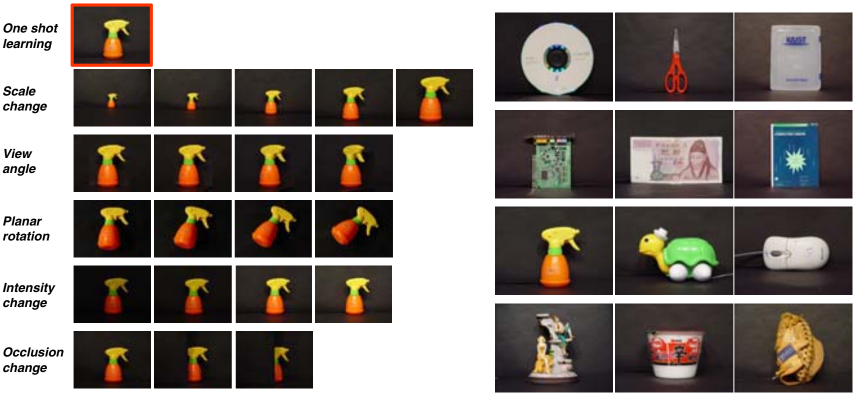


Fig. 6. We use frontal 104 views for DB (right) generation and 20 test set (5 scales, 4 view angles, 4 rotation, 4 intensity change, 3 occlusions) per object (left)

as COIL-100 and Caltech DB, we evaluate the proposed method using our own database because our research goal is to measure the properties of features in terms of scale change, view angle change, planar rotation, illumination intensity change, and occlusion. The total number of objects is 104: related test images are shown in Fig. 6. These DB and test images are acquired using a SONY F717 digital camera and resized to 320×240.

We compare the performance of the proposed G-RIF with SIFT, the state-of-the art feature [3]. We evaluate the features using the nearest neighbor classifier with direct voting (NNC-voting) which is used commonly in local feature-based object recognition approaches. NNC-based voting is a very similar concept to

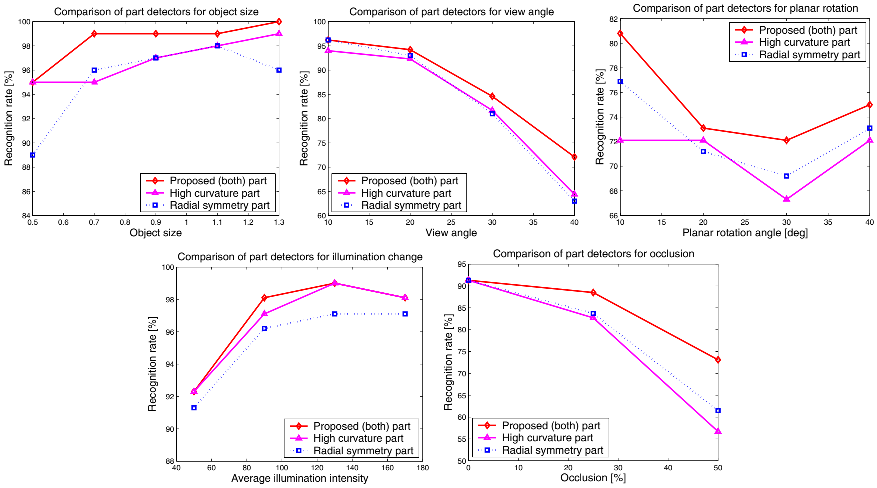


Fig. 7. Evaluation of part detectors (radial symmetry part, high curvature part, perceptual part): The proposed part detector shows the best performance for all test set

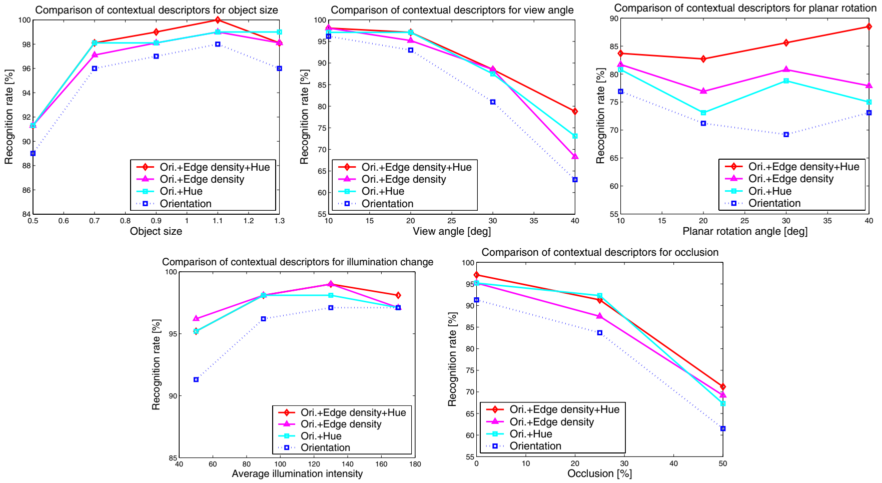


Fig. 8. Evaluation of part descriptors (ori. only, ori+hue, ori+edge, ori+hue+edge): The full descriptor shows almost best performance

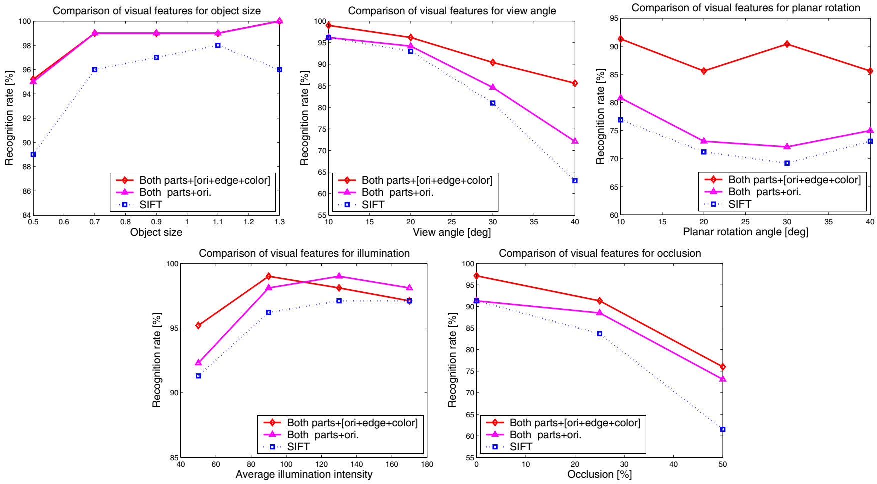


Fig. 9. Summary of visual features (SIFT, perceptual part + edge orientation, G-RIF): G-RIF shows the best performance. Both parts+ori and SIFT follow G-RIF.

the winner-take-all (WTA). We use the binary program offered by Lowe [3] for the accurate comparison.

Fig. 7 shows the performance of the proposed perceptual part detectors. We use the same descriptor (edge orientation only) with the same Euclidean distance threshold (0.2) used in NNC-based voting. The proposed perceptual part detector outperforms single part detectors in most test sets. The maximal recognition rate is higher than the part detector of SIFT (radial symmetry part) by

15%. Fig. 8 shows the performance of the proposed part descriptor with the SIFT descriptor. We used the same radial symmetry part detector to show the power of descriptors only. The full contextual descriptor (edge orientation + edge density + hue field) shows the best performance except the illumination intensity change set. In this case, the performance is fair compared to the other contextual descriptors. Under severe illumination intensity change or different light sources, it is reasonable to use the contextual descriptor of edge orientation with edge density. Fig. 9 summarizes the performance of the SIFT, perceptual part detector with edge orientation descriptor, and the G-RIF (both parts with general descriptor). The G-RIF always outperforms the SIFT in all test sets. This good performance is originates from the effective use of image structures (radial symmetry point with a high curvature point) of the proposed visual part detector and effective spatial coding of multiple features in a unified way. The dimension of G-RIF is 189 (21*4+21*4+21) and that of SIFT is 128 (16*8). The average extraction time of G-RIF is 0.15sec and that of SIFT is 0.11sec in a 320*240 image under AMD 2400+. This difference is due to the number of visual parts. The G-RIF extracts twice number of parts than the SIFT does.

6 Conclusions

In this paper, we introduced a Generalized-Robust Invariant Feature which shows good performance in terms of selectivity (recognition accuracy) and invariance (to various test images). First, we detect perceptually meaningful visual parts derived from the properties of the visual receptive field of V1. Applying the Tune-MAX scheme to two basis Gabor kernels can extract complementary visual parts (Fig. 4). Second, we also proposed a generalized contextual encoding scheme based on the properties of receptive field V4 and attention of the HVS. The information of edge field and hue field is characterized by the localized histogram weighted according to the attentional strength. It is a generalized form of SIFT descriptor, shape context, minutia descriptor. The performance of the G-RIF compared with the state-of-the art feature shows the recognition power using the NNC-based simple voting. Effective utilization of image structures and pixel information give good performance in feature-based object recognition.

Acknowledgements

This research has been supported by the Korean Ministry of Science and Technology for National Research Laboratory Program (M1-0302-00-0064), Korea.

References

1. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. IJCV **37** (2000) 151–172
2. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. IJCV **60** (2004) 63–86

3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
4. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI **27** (2005) 1615–1630
5. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: CVPR'04. (2004) 506–513
6. Tico, M., Kuosmanen, P.: Fingerprint matching using an orientation-based minutia descriptor. PAMI **25** (2003) 1009–1014
7. Belongie, S., Malik, J., Puzicha, J.: Shape matching and bject recognition using shape contexts. PAMI **24** (2002) 509–522
8. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: CVPR'05. (2005)
9. Ringach, D.L.: Spatial structure and symmetry of simple-cell receptive field in macaque primary visual cortex. J. Neurophysiol. **88** (2001) 455–463
10. Gawne, T.J., Martin, J.M.: Responses of primate visual cortical V4 neurons to simultaneously presented stimuli. J. Neurophysiol. **88** (2002) 1128–1135
11. Boynton, G.M.: Adaptation and attentional selection. Nature Neurosci. **7** (2004) 8–10
12. Chomat, O., de Verdière, V.C., Hall, D., Crowley, J.L.: Local scale selection for gaussian based description techniques. In: ECCV'00, London, UK, Springer-Verlag (2000) 117–133
13. Wurtz, R.H., Kandel, E.R.: Perception of motion, depth and form. Principles of Neural Science (2000) 548–571
14. Kouh, M., Riesenhuber, M.: Investigating shape representation in area V4 with hmax: Orientation and grating selectivitiesn. (Technical report)
15. Biederman, I.: Recognition-by-component: A theory of human image understanding. Psychological Review **94** (1987) 115–147
16. Okada, K., Comaniciu, D.: Scale selection for anisotropic scale-space: Application to volumetric tumor characterization. In: CVPR'04. (2004) 594–601
17. Pasupathy, A., Connor, C.E.: Shape representation in area V4: Position-specific tuning for boundary conformation. J. Neurophysiol **86** (2001) 2505–2519
18. Reisfeld, D., Wolfson, H., Yeshurun, Y.: Context-free attentional operators: The generalized symmetry transform. IJCV **14** (1995) 119–130
19. Latecki, L.J., Lakämper, R.: Convexity rule for shape decomposition based on discrete contour evolution. CVIU **73** (1999) 441–454
20. Loy, G., Zelinsky, A.: Fast radial symmetry transform for detecting points of interest. PAMI **25** (2003) 959–973
21. Schiele, B., Crowley, J.L.: Recognition without correspondence using multidimensional receptive field histograms. IJCV **36** (2000) 31–50

A Framework for 3D Object Recognition Using the Kernel Constrained Mutual Subspace Method

Kazuhiro Fukui[1], Björn Stenger[2], and Osamu Yamaguchi[2]

[1] Graduate School of Systems and Information Engineering, University of Tsukuba
kfukui@cs.tsukuba.ac.jp
[2] Corporate Research and Development Center, Toshiba corporation
bjorn@eel.rdc.toshiba.co.jp, osamu1.yamaguchi@toshiba.co.jp

Abstract. This paper introduces the kernel constrained mutual subspace method (KCMSM) and provides a new framework for 3D object recognition by applying it to multiple view images. KCMSM is a kernel method for classifying a set of patterns. An input pattern $\mathbf{x}$ is mapped into the high-dimensional feature space $\mathcal{F}$ via a nonlinear function ϕ, and the mapped pattern $\phi(\mathbf{x})$ is projected onto the kernel generalized difference subspace, which represents the difference among subspaces in the feature space $\mathcal{F}$. KCMSM classifies an input set based on the canonical angles between the input subspace and a reference subspace. This subspace is generated from the mapped patterns on the kernel generalized difference subspace, using principal component analysis. This framework is similar to conventional kernel methods using canonical angles, however, the method is different in that it includes a powerful feature extraction step for the classification of the subspaces in the feature space $\mathcal{F}$ by projecting the data onto the kernel generalized difference subspace. The validity of our method is demonstrated by experiments in a 3D object recognition task using multiview images.

1 Introduction

This paper introduces the kernel constrained mutual subspace Method (KCMSM), which provides a new framework for view-based 3D object recognition.

Many view-based methods have been proposed to achieve high-performance object recognition. Of these, the mutual subspace method (MSM)[2] with the ability of handling multiple images, such as sequential images, and multiview images, is one of the most suitable and efficient methods for object recognition. Let an $n \times n$ pixel pattern be treated as a vector $\mathbf{x}$ in n^2-dimensional space (called input space $\mathcal{I}$). In MSM, the set of patterns $\{\mathbf{x}\}$ of each class is represented by a low-dimensional linear subspace using Karhunen-Loève (KL) expansion, also known as principal component analysis (PCA). The classification of a set of patterns is executed based on the canonical angles θ_i between subspaces as shown in Fig.1, where smaller angles indicate higher similarity between two subspaces.

MSM works well when the distribution of each class can be represented by a linear subspace with no overlap of the distributions. However, this representation

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 315–324, 2006.

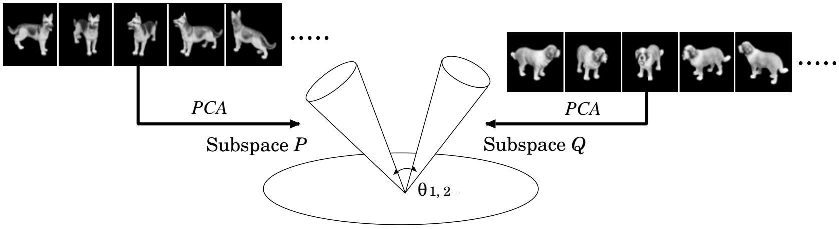


Fig. 1. Measuring the similarity between two distributions of view patterns with canonical angles $\theta_{1,2,\ldots}$

is not suitable for representing highly nonlinear structures, such as those of multiview patterns of a 3D object.

To overcome this problem, MSM has been extended to nonlinear *kernel MSM* (KMSM)[4, 5] using the "kernel trick" [3]. An input pattern $\mathbf{x}$ is mapped onto the very high dimensional (in some cases infinite) feature space $\mathcal{F}$ via a nonlinear map ϕ. Then, MSM is applied to the linear subspaces generated from the mapped patterns $\{\phi(\mathbf{x})\}$, where the linear subspace in the feature space $\mathcal{F}$ is a nonlinear subspace as seen from the input space $\mathcal{I}$. The kernel MSM has better performance compared to MSM, since the distribution of the mapped patterns $\{\phi(\mathbf{x})\}$ can be represented by a subspace in the feature space $\mathcal{F}$ without overlapping of distributions. However, in practice the classification performance KMSM is still insufficient for many applications as is the case with other methods based on PCA, because the subspaces are generated independently of each other [1]. Although each subspace represents the distribution of the training patterns well in terms of a least mean square approximation, there is no reason to assume a priori that it is the optimal subspace in terms of classification performance.

This issue is addressed by the *constrained MSM* (CMSM)[6]. CMSM performs the MSM algorithm on the patterns after projecting them onto the generalized difference subspace $\mathcal{D}$ (called difference subspace), wherein the differences among subspaces are contained, as shown in Fig.2. CMSM has significantly higher classification performance compared to MSM since it selectively uses the canonical angles θ_d calculated from discriminative features extracted by the projection[6, 7].

The idea in this paper is to incorporate the mechanism of this powerful feature extraction of the constrained MSM into the kernel MSM: We construct the generalized difference subspace in the feature space $\mathcal{F}$, and project the mapped pattern $\phi(\mathbf{x})$ onto this subspace for kernel MSM. This projection τ can be regarded as an effective nonlinear feature extraction step for classification of the subspaces, as seen from the input space $\mathcal{I}$. We name the difference subspace in the feature space the *nonlinear kernel generalized difference subspace* $\mathcal{D}_\phi$ and the KMSM with the projection the *kernel CMSM* (KCMSM). One question that arises is how to calculate the projection onto the difference subspace. We show that it is in fact possible to calculate the projection using the kernel trick, because it consists of the inner products. Consequently, KCMSM carries out the MSM algorithm on the extracted feature patterns $\{\tau(\phi(\mathbf{x}))\}$ by the projection.

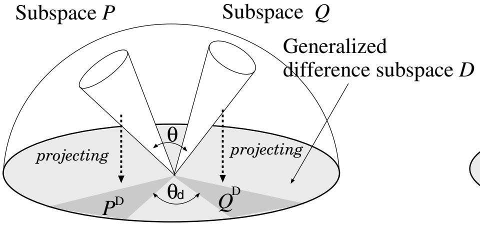


Fig. 2. Concept of CMSM

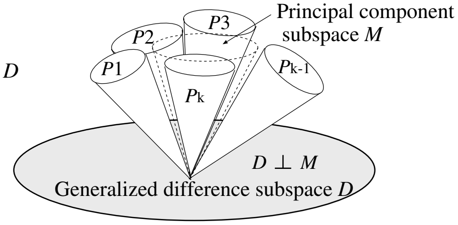

Principal component subspace M
P1
P2
P3
Pk
Pk-1
$D \perp M$
Generalized difference subspace D

Fig. 3. Generalized difference subspace

In addition to the high classification performance, our method has also an ability of handling multiple classes in a simple framework. This is indispensable for many applications of object recognition, such as face recognition. On the other hand, many other types of kernel methods do not have this ability. For instance, the well-known support vector machine classifier is basically a two-class classifier[10]. Thus, the classification process becomes more involved and time-consuming in a multiple class problem.

This paper is organized as follows. In section 2, we review the CMSM algorithm. In section 3, we introduce the kernel generalized difference subspace in the feature space, and construct KCMSM. Our method is demonstrated by the evaluation experiments in section 4. In Section 5, conclusions are presented.

2 Recognition Based on CMSM

In this section, we first review the concepts of the canonical angle and the generalized difference subspace. Then, we explain the CMSM algorithm.

2.1 Calculation of Canonical Angles

A natural way for comparing two subspaces is by computing the *canonical angles* between them [8]. We can obtain N canonical angles θ_i (for convenience $N \leq M$) between an M-dimensional input subspace $\mathcal{P}$ and an N-dimensional reference subspace $\mathcal{Q}$ in the f-dimensional input space $\mathcal{I}$. Let $\mathbf{\Phi}_i$ and $\mathbf{\Psi}_i$ denote the i-th f-dimensional orthonormal basis vectors of the subspaces $\mathcal{P}$ and $\mathcal{Q}$, respectively. The value $\cos^2\theta_i$ of the i-th smallest canonical angle θ_i $(i = 1, \ldots, N)$ is obtained as the i-th largest eigenvalue λ_i of the following $N \times N$ matrix $\mathbf{X}$[6, 8]:

$$\mathbf{X}\mathbf{c} = \lambda\mathbf{c}, \tag{1}$$

$$\mathbf{X} = (x_{ij}),\ x_{ij} = \textstyle\sum_{k=1}^{M}(\mathbf{\Psi}_i \cdot \mathbf{\Phi}_k)(\mathbf{\Phi}_k \cdot \mathbf{\Psi}_j).$$

2.2 Generation of the Generalized Difference Subspace

The generalized difference subspace represents the difference among multiple $k(\geq 2)$ subspaces as an extension of the difference subspace defined as the difference between two subspaces.

Given $k(\geq 2)$ N-dimensional subspaces, the generalized difference subspace $\mathcal{D}$ is defined as the subspace which results by removing the principal component subspace $\mathcal{M}$ of all subspaces from the sum subspace $\mathcal{S}$ of these subspaces as shown in Fig.3. According to this definition, $\mathcal{D}$ is spanned by N_d eigenvectors $\mathbf{d}_i(i = N\times k - N_d, \ldots, N\times k)$ corresponding to the N_d smallest eigenvalues, of the matrix $\mathbf{G} = \sum_{i=1}^{k} \mathbf{P}_i$ of projection matrices $\mathbf{P}_i$. where the projection matrix $\mathbf{P}_i = \sum_{j=1}^{N} \mathbf{\Phi}_j^i \mathbf{\Phi}_j^{i\top}$, $\mathbf{\Phi}_j^i$ is the j-th orthonormal basis vector of the i-th class subspace. The eigenvectors, $\mathbf{d}_i$ correspond to the i-th eigenvalue λ_i in descending order.

The projection onto the generalized difference subspace $\mathcal{D}$ corresponds to removing the principal (common) component subspace $\mathcal{M}$ from the sum subspace $\mathcal{S}$. This projection has the effect of expanding the canonical angles between subspaces and forms a relation between subspaces which is close to the orthogonal relation, thus improving the performance of classification based on canonical angles [6].

2.3 The CMSM Algorithm

The steps of the CMSM algorithm are as follows:

1. The reference subspace $\mathcal{P}_k^D$ of each class k is generated from the training patterns projected onto the generalized difference subspace $\mathcal{D}$ using PCA.
2. The input subspace $\mathcal{P}_{in}^D$ is generated from the input test patterns projected onto $\mathcal{D}$ using PCA.
3. The canonical angles θ between the $\mathcal{P}_{in}^D$ and the $\mathcal{P}_k^D$ of each class are calculated using Eq.(1).
4. The similarity $S[t]$ is calculated as the mean value $\frac{1}{t}\sum_{i=1}^{t}\cos^2\theta_i$. The reference subspace with the highest similarity is determined to be that of the identified class, given the similarity is above a threshold.

Instead of steps 0 and 1, we can also obtain the canonical angles by the procedure described in [6]. In this method, the input subspace and the reference subspaces are first generated from the set of patterns, and then these generated subspaces are projected onto $\mathcal{D}$.

3 The Kernel Constrained Mutual Subspace Method

In this section, we first review kernel Principal Component Analysis (KPCA). Next, we define the kernel generalized difference subspace using the technique of the kernel PCA, and we describe the new KCMSM algorithm.

3.1 Kernel PCA

The nonlinear function ϕ maps the patterns $\mathbf{x} = (x_1, \ldots, x_f)^\top$ of an f-dimensional input space $\mathcal{I}$ onto an f_ϕ-dimensional feature space $\mathcal{F}$: $\phi : R^f \rightarrow R^{f_\phi}$, $\mathbf{x} \rightarrow \phi(\mathbf{x})$. To perform PCA on the mapped patterns, we need to calculate the inner product

$(\phi(\mathbf{x}) \cdot \phi(\mathbf{y}))$ between the function values. However, this calculation is difficult, because the dimension of the feature space $\mathcal{F}$ can be very high, possibly infinite. However, if the nonlinear map ϕ is defined through a kernel function $k(\mathbf{x}, \mathbf{y})$ which satisfies Mercer's conditions, the inner products $(\phi(\mathbf{x}) \cdot \phi(\mathbf{y}))$ can be calculated from the inner products $(\mathbf{x} \cdot \mathbf{y})$. This technique is known as the "kernel trick". A common choice is to use the Gaussian kernel function[3]:

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{\sigma^2}\right) \ . \tag{2}$$

The function ϕ maps an input pattern onto an infinite feature space $\mathcal{F}$. The PCA of the mapped patterns is called kernel PCA[3], and the linear subspace generated by the kernel PCA are nonlinear subspaces in the input space $\mathcal{I}$.

Given the N-dimensional nonlinear subspace $\mathcal{V}_k$ of class k generated from m training patterns $\mathbf{x}_i, (i = 1, \ldots, m)$, the N orthonormal basis vectors $\mathbf{e}_i^k$,$(i = 1, \ldots, N)$, which span the nonlinear subspace $\mathcal{V}_k$, can be represented by the linear combination of the m $\phi(\mathbf{x}_i^k)$,$(i = 1, \ldots, m)$ as follows

$$\mathbf{e}_i^k = \sum_{j=1}^{m} \mathrm{a}_{ij}^k \, \phi(\mathbf{x}_j^k), \tag{3}$$

where the coefficient a_{ij} is the j-th component of the eigenvector $\mathbf{a}_i$ corresponding to the i-th largest eigenvalue λ_i of the $m \times m$ matrix $\mathbf{K}$ defined by the following equation:

$$\begin{aligned} \mathbf{Ka} &= \lambda \mathbf{a} \\ \mathrm{k}_{ij} &= (\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)) \\ &= k(\mathbf{x}_i, \mathbf{x}_j), \end{aligned} \tag{4}$$

where $\mathbf{a}_i$ is normalized to satisfy $\lambda_i(\mathbf{a}_i \cdot \mathbf{a}_i)$=1. We can compute the projection of the mapped $\phi(\mathbf{x})$ onto the i-th orthonormal basis vector $\mathbf{e}_i^k$ of the nonlinear subspace of class k by the following equation:

$$(\phi(\mathbf{x}), \mathbf{e}_i^k) = \sum_{j=1}^{m} \mathrm{a}_{ij}^k \, k(\mathbf{x}, \mathbf{x}_j). \tag{5}$$

3.2 Generation of the Kernel Difference Subspace

It is possible to compute the projection of a mapped pattern $\phi(\mathbf{x})$ onto the kernel generalized difference subspace $\mathcal{D}^{\phi}$ using the kernel trick, since it consists of the inner products in the feature space $\mathcal{F}$. Let the N_d^{ϕ}-dimensional $\mathcal{D}^{\phi}$ be generated from the r N-dimensional nonlinear subspace $\mathcal{V}_k, (k = 1, \ldots, r)$. Firstly we calculate the orthonormal bases of kernel generalized difference subspace from all the orthonormal basis vectors of r nonlinear subspaces, namely, $r \times N$ basis vectors. This calculation corresponds to the PCA of all basis vectors. Define $\mathbf{E}$ to be a matrix, which contains all basis vectors as columns:

$$\mathbf{E} = [\mathbf{e}_1^1, \ldots, \mathbf{e}_N^1, \ldots, \mathbf{e}_1^r, \ldots, \mathbf{e}_N^r]. \tag{6}$$

Secondly we solve the eigenvalue problem of the matrix $\mathbf{D}$ defined by the following equation:

$$\mathbf{D}\mathbf{b} = \beta\mathbf{b} \tag{7}$$

$$\mathrm{D}_{ij} = (\mathbf{E}[i] \cdot \mathbf{E}[j]), \quad (i,j = 1,\ldots,r{\times}N) \ , \tag{8}$$

where $\mathbf{E}[i]$ represents the i-th column of the matrix $\mathbf{E}$.

The inner product between the i-th orthonormal basis vector $\mathbf{e}_i^k$ of the class k subspace and the j-th orthonormal basis vector $\mathbf{e}_j^{k^*}$ of the class k^* subspace can be obtained as the linear combination of kernel functions $k(\mathbf{x}^k, \mathbf{x}^{k^*})$ as follows:

$$(\mathbf{e}_i^k \cdot \mathbf{e}_j^{k^*}) = (\sum_{s=1}^{m} \mathrm{a}_{is}^k \phi(\mathbf{x}_s) \cdot \sum_{t=1}^{m} \mathrm{a}_{jt}^{k^*} \phi(\mathbf{x}_t^*)) \tag{9}$$

$$= \sum_{s=1}^{m} \sum_{t=1}^{m} \mathrm{a}_{is}^k \mathrm{a}_{jt}^{k^*} (\phi(\mathbf{x}_s) \cdot \phi(\mathbf{x}_t^*)) \tag{10}$$

$$= \sum_{s=1}^{m} \sum_{t=1}^{m} \mathrm{a}_{is}^k \mathrm{a}_{jt}^{k^*} k(\mathbf{x}_s, \mathbf{x}_t^*) \tag{11}$$

The i-th orthonormal basis vector $\mathbf{d}_i^\phi$ of the kernel generalized difference subspace $\mathcal{D}^\phi$ can be represented by a linear combination of the vectors $\mathbf{E}[j]$ $(j = 1,\ldots,r{\times}N)$, $\mathbf{d}_i^\phi = \sum_{j=1}^{r\times N} \mathrm{b}_{ij}\mathbf{E}[j]$, where the weighting coefficient b_{ij} is the j-th component of the eigenvector $\mathbf{b}_i$ corresponding to the i-th smallest eigenvalue β_i of matrix $\mathbf{D}$ under the condition that the vector $\mathbf{b}_i$ is normalized to satisfy that $\beta_i(\mathbf{b}_i \cdot \mathbf{b}_i)$=1.

Let $\mathbf{E}[j]$ denote the $\eta(j)$-th basic vector of class $\zeta(j)$. The above orthonormal basis vector $\mathbf{d}_i^\phi$ is converted to the following equation:

$$\sum_{j=1}^{r\times N} \mathrm{b}_{ij}\mathbf{E}[j] = \sum_{j=1}^{r\times N} \mathrm{b}_{ij} \sum_{s=1}^{m} \mathrm{a}_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)}) \tag{12}$$

$$= \sum_{j=1}^{r\times N} \sum_{s=1}^{m} \mathrm{b}_{ij} \mathrm{a}_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)}) \ . \tag{13}$$

3.3 Projection onto the Kernel Difference Subspace

Although it is impossible to calculate the orthonormal basis vector $\mathbf{d}_i^\phi$ of the kernel generalized difference subspace $\mathcal{D}^\phi$, the projection of the mapped pattern $\phi(\mathbf{x})$ onto this vector $\mathbf{d}_i^\phi$ can be calculated from an input pattern $\mathbf{x}$ and all $m{\times}r$ training patterns $\mathbf{x}_s^k(s = 1,\ldots,m, k = 1,\ldots,r)$.

$$(\phi(\mathbf{x}) \cdot \mathbf{d}_i^\phi) = \sum_{j=1}^{r\times N} \sum_{s=1}^{m} \mathrm{b}_{ij} \mathrm{a}_{\eta(j)s}^{\zeta(j)} (\phi(\mathbf{x}_s^{\zeta(j)}) \cdot \phi(\mathbf{x})) \tag{14}$$

$$= \sum_{j=1}^{r\times N} \sum_{s=1}^{m} \mathrm{b}_{ij} \mathrm{a}_{\eta(j)s}^{\zeta(j)} k(\mathbf{x}_s^{\zeta(j)}, \mathbf{x}) \tag{15}$$

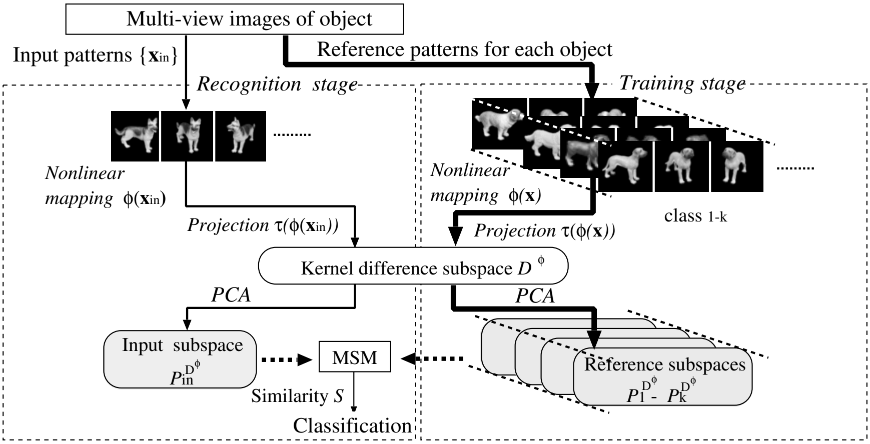


Fig. 4. Flow of object recognition using KCMSM

Note that we can compute $k(\mathbf{x}_s^{\zeta(j)}, \mathbf{x})$ through Eq.(2) easily. Finally, each component of the projection $\tau(\phi(\mathbf{x}))$ of the mapped $\phi(\mathbf{x})$ onto the $N_d^\phi(< r{\times}N)$-dimensional kernel generalized difference subspace is represented as the following: $\tau(\phi(\mathbf{x})) = (z_1,\ z_2, \ldots,\ z_{N_d^\phi})^\top,\ z_i = (\phi(\mathbf{x}) \cdot \mathbf{d}_i^\phi)$.

3.4 The KCMSM Algorithm

We construct KCMSM by applying linear MSM to the projection $\tau(\phi(\mathbf{x}))$. Fig.4 shows a schematic of the KCMSM algorithm.

In the training stage, the mapped patterns $\phi(\mathbf{x}_{ki})$ of the patterns $\mathbf{x}_i^k$, $(i = 1, \ldots, m)$ belonging to class k, are projected onto the kernel difference subspace $\mathcal{D}^\phi$. Then, the N_ϕ-dimensional linear reference subspace $\mathcal{P}_k^{D^\phi}$ of each class k is generated from the mapped patterns $\tau(\phi(\mathbf{x}_i^k))$ using PCA.

In the recognition stage, we generate the linear input subspace $\mathcal{P}_{in}^{D^\phi}$ on the $\mathcal{D}^\phi$ from the input patterns $\mathbf{x}_i$,$(i = 1, \ldots, m)$. Then we compute the similarity S, defined in Sec.2.3, between the input subspace $\mathcal{P}_{in}^{D^\phi}$ and each reference subspace $\mathcal{P}_k^{D^\phi}$. Finally the object class is determined as the reference subspace with the highest similarity S, given that S is above a threshold value.

4 Evaluation Experiments

We compared KCMSM with MSM, CMSM, and KMSM using the public database of the multi-view image set (ETH-80: Cropped-close128)[9].

Experimental conditions: We selected 30 similar models (10 of each; cows, dogs, and horses) from the database as shown in Fig.5(a) and used them for the evaluation. The images of each model were captured from 41 views as shown in Fig.5(b). The view directions are the same for all models. All images are cropped, so that they contain only the object without any border area.

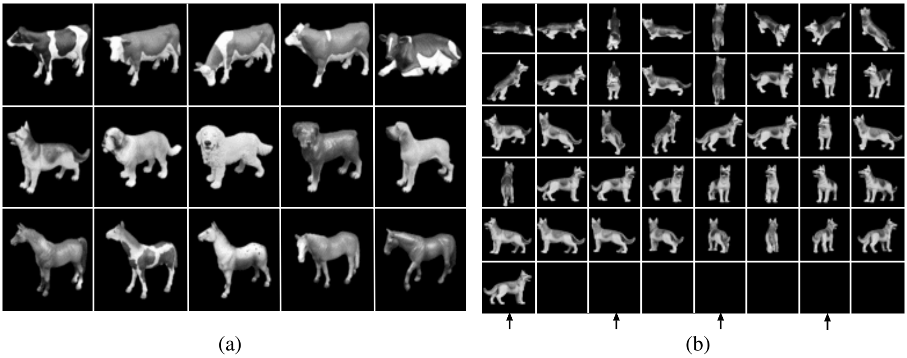


Fig. 5. Data set: (a) Subset of the input images, Top: cows, Middle: dogs, Bottom: horses. (b) All 41 view-patterns of a dog model: the columns indicated by the arrows are used as the training data.

The odd numbered images (21 frames) and the even numbered images (20 frames) were used for training and evaluation, respectively. We prepared 10 datasets for each model by making the start frame number i change from 1 to 10 where 10 frames from i-th frame to $i + 9$-th is one set. The total number of the evaluation trials is 9000(=10×30×30). The evaluation was performed using measures for recognition rate and separability: a normalized index of classification ability. Given two classes of similarities within a model category and similarities across different model category, separability was calculated as a ratio of the between-class scatter to the total scatter.

We converted the 180×180 pixels color images to 15×15 pixels monochrome images and use them as the evaluation data. Thus, the dimension f of a pattern is 225(=15×15). The dimensions of the input subspace and the reference subspaces were set to 7 in all methods.

$\mathcal{P}^D_{in}$ and $\mathcal{P}^D_k$ were generated from the patterns projected on the generalized difference subspace. The difference subspace $\mathcal{D}$ was generated from thirty 20-dimensional subspaces of all classes according to the procedure described in Sec. 2.2. We varied the dimension N_d of $\mathcal{D}$ between 190 and 215 to compare the performance. The kernel difference subspace $\mathcal{D}_\phi$ was generated from thirty 20-dimensional subspaces of all classes according to the procedure described in Sec.2.3. We varied the dimension N^ϕ_d of $\mathcal{D}_\phi$ between 100 and 550. We used a Gaussian kernel with $\sigma^2 = 0.05$ defined by Eq.(2).

Experimental results: Table 1 shows the recognition rate and the separability. In the tables, the notation *method type – dimension of the difference subspace* is used and t denotes the number of the canonical angles used for the similarity $S[\mathrm{t}]$ defined in step 3 of Section 2.3.

From these results, it can be observed that the performance of the nonlinear methods (KMSM and KCMSM) is superior to the one of the linear methods (MSM and CMSM), indicating that the recognition of multiple view images is typically a nonlinear problem.

Table 1. Performance of each method

(a) Recognition rate (%)

	t=1	t=2	t=3	t=4
MSM	72.7	73.7	**76.3**	74.3
CMSM-215	75.7	**81.3**	76.3	73.7
CMSM-200	73.3	81.0	79.3	77.7
CMSM-190	71.0	73.0	73.0	75.0
KMSM	84.7	**87.0**	82.0	81.7
KCMSM-550	83.0	85.3	85.7	86.3
KCMSM-500	79.3	85.0	87.0	87.0
KCMSM-450	82.0	88.0	89.3	**89.7**
KCMSM-400	83.3	87.7	88.3	89.7
KCMSM-300	81.0	87.7	88.7	89.0
KCMSM-200	81.7	81.7	83.3	83.3
KCMSM-100	57.7	62.7	68.0	65.3

(b) Separability

	t=1	t=2	t=3	t=4
MSM	0.055	0.074	**0.082**	0.080
CMSM-215	0.203	0.236	0.242	0.236
CMSM-200	0.215	**0.257**	0.254	0.245
CMSM-190	0.229	0.255	0.249	0.244
KMSM	0.375	0.420	0.420	**0.429**
KCMSM-550	0.538	0.581	0.584	0.538
KCMSM-500	0.556	0.607	0.616	0.612
KCMSM-450	0.549	0.618	0.621	**0.621**
KCMSM-400	0.529	0.601	0.607	0.609
KCMSM-300	0.483	0.536	0.545	0.545
KCMSM-200	0.340	0.385	0.403	0.408
KCMSM-100	0.141	0.194	0.212	0.213

The performance of MSM was improved by the nonlinear extension of MSM to KMSM where the recognition rate increased from 76.3% to 87.0% and the separability increased from 0.082 to 0.429.

The new KCMSM improved the recognition rate further to 89.7% and increased the separability by a value of almost 0.2 in comparison to KMSM. This confirms the effectiveness of projection the onto the kernel difference subspace, which serves as a feature extraction step in the feature space $\mathcal{F}$. In particular, the high separability of KCMSM is remarkable. This indicates that KCMSM can maintain high performance even if the number of classes becomes larger.

The classification ability of KCMSM was improved while increasing t of the similarity $S[t]$. These results show that the similarity $S[1]$ is not sufficient for classification of the models with similar 3D shapes. This is because $S[1]$ utilizes only the information of a single view. On the other hand, $S[t](t{\geq}2)$ reflects the information of 3D shape including multiple views. Note that the recognition rate of KMSM decreased, although it is also a nonlinear method. From this, one can deduce that the projection onto the kernel difference subspace ensures the validity of the similarity $S[t],(t{\geq}2)$.

In comparison between KCMSM-450 and KCMSM-300, the extreme degradation of performance does not appear even when the dimension of the kernel difference subspace decreased to 300. This implies that we can decrease the dimension N_d^{ϕ} within the permissible range to reduce the computing cost.

5 Summary and Conclusions

This paper has introduced the kernel constrained mutual subspace method (KCMSM) and demonstrated its application to 3D object recognition. We showed a significant performance improvement over kernel MSM, which is a state-of-the-art method for classifying multiple view patterns with nonlinear structure. The projection onto the kernel generalized difference subspace can be viewed as a nonlinear feature extraction step based on the concept of constrained MSM. The extracted features by this projection could improve the classification ability of kernel MSM. The validity of KCMSM was shown through the experimental results with the set of the multiple view patterns of 3D objects.

In future work, we will evaluate the performance of KCMSM using other databases, such as a face image database. In this case, the comparisons with other kernel methods[10] are required. Another problem that remains to be addressed is the computation of the eigen–problems of the matrices **K** and **D**, which becomes difficult when the size of these matrices become large in proportion to the numbers of the classes and the training patterns. To solve this problem, the reduction of the number of the training patterns is most effective. Thus, the framework of ensemble learning[11] is useful, since it can obtain high performance using only a few training patterns.

References

1. Oja, E.: Subspace methods of pattern recognition. Research Studies Press, England, (1983)
2. Yamaguchi, O., Fukui, K., and Maeda, K.: Face recognition using temporal image sequence. Proc. Third International Conference on Automatic Face and Gesture Recognition, (1998) 318–323
3. Schölkopf, B., Smola, A. and Müller, K.-R.: Nonlinear principal component analysis as a kernel eigenvalue problem. Neural Computation, vol. 10, (1998) 1299–1319
4. Sakano, H., and Mukawa, N.: Kernel mutual subspace method for robust facial image recognition. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES2000) (2000) 245–248
5. Wolf, L. and Shashua, A.: Kernel principal angles for classification machines with applications to image sequence interpretation. Proc. CVPR, (2003) 635–642
6. Fukui, K. and Yamaguchi, O.: Face recognition using multi-viewpoint patterns for robot vision. 11th International Symposium of Robotics Research (ISRR'03), Springer, (2003) 192–201
7. Arandjelović, O., Shakhnarovich, G., Fisher, J., Cipolla, R. and Darrell, T.: Face recognition with image sets using manifold density divergence. Proc. CVPR, vol.1, (2005) 581–588
8. Chatelin, F.: Eigenvalues of matrices. John Wiley & Sons, Chichester, (1993)
9. Leibe, B. and Schiele, B.: Analyzing appearance and contour based methods for object categorization. Proc. CVPR, (2003) 409–415
10. Schölkopf, B. and Smola, A.: Learning with Kernels. The MIT Press, (1999)
11. Breiman, L.: Bagging Predictors. Machine Learning, vol.24, no.2 (1996) 123–140

An Iterative Method for Preserving Edges and Reducing Noise in High Resolution Image Reconstruction

Chanho Jung and Gyeonghwan Kim

Department of Electronic Engineering, Sogang University,
C.P.O Box 1142, Seoul 100-611, Rep. of Korea
{97peter, gkim}@sogang.ac.kr

Abstract. In this paper we present simple iterative method for obtaining high resolution images with enhanced edges but reduced noise. In the method the trade off between the output noise and the edge preservation is being taken care of by employing an energy-based framework. In each iteration, two processes are involved: 1) the edge enhancement and reducing noise which occurs during the edge enhancement process, and 2) consideration of the fidelity to the low resolution images and the smoothness constraint of the restored high resolution image. In the implementation, the first process is designed to be embedded into the second process. And a termination condition is established by taking into account high frequency energy of the image being restored and error energy for each low resolution image. Experimental results show that the proposed method produces high resolution images in which edges are preserved with reduced noise, comparing to the ones produced by conventional methods. Moreover, it turns out that the approach is less sensitive to initialization factor in terms of PSNR and subjective visual quality.

1 Introduction

Advent of new types of device for image display and storage has created interests on the high resolution image reconstruction from multiple low resolution images, including video standard conversion, multimedia imaging, and image analysis [1, 2].

In order to deal with this typical ill-posed problem, various optimization schemes, such as Bayesian maximum a posteriori (MAP) estimation, projection onto convex sets (POCS) algorithm, iterative back projection (IBP) approach, and constrained least square (CLS) algorithm have been widely applied [3]. In the MAP estimation and the CLS algorithm, the optimal high resolution image can be obtained by iteratively minimizing the objective function defined. During the process, edges of the restored image are gradually sharpened along with noise components, as high frequency components are restored [4]. Nevertheless, smoothing effect occurring in edges is unavoidable as we consider the smoothness constraint. Also, it is quite common to observe typical artifacts, such as staircase

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 325–334, 2006.

effect, while the reconstruction is performed. In terms of edge preserving, a prior assumption using Huber-Markov random field (HMRF) model and the spatial adaptive algorithm using local properties such as local mean and local variance have been employed [5, 6].

In this paper, to alleviate the problems, we propose a simple iterative method for obtaining high resolution images with enhanced edges but reduced noise. In the proposed method, the iteration includes two processes: one is for enhancing edges and reducing noise which is being amplified while the edges are enhanced, and the other is for maintaining the fidelity and the smoothness constraints. Note that the result of the edge enhancement process is effectively controlled by the application of a smoothness constraint. As the iteration includes the contradictory processes in terms of preserving edges, each process needs to be well balanced as the iteration continues.

In the former process, a sharpening filter is employed for edge enhancement in the restored image. The weight of the filter is determined depending on the noise components in the image being restored. Possible enhancement of noise is managed by employing a median filter conditionally. In the later process, the objective function which satisfies the constraints is obtained in each iteration and the restored high resolution image is created based on the objective function. The termination condition of the iteration is set by considering a sum of high frequency energy of the image being restored and error energy for each low resolution image.

The rest of the paper is organized as follows: In section 2, a general observation model is briefly introduced. In section 3, the proposed iterative method for high resolution image reconstruction is presented. Experimental results are shown in section 4, and finally conclusion is given in section 5.

2 Observation Model

We assume that a high resolution image is degraded by motion between neighboring frames, blurring and downsampling caused by sensor, and noise occurred during the acquisition. In the model defined in (1), let us assume that a high resolution image is restored using p distorted images.

$$\mathbf{y}_k = \mathbf{A}_k\mathbf{x} + \mathbf{n}_k \text{ for } k = l - \tfrac{p-1}{2}, \ldots, l, \ldots, l + \tfrac{p-1}{2} \tag{1}$$

where $\mathbf{y}_k$ and $\mathbf{n}_k$ are k^{th} observed image and the additive zero mean Gaussian noise for k^{th} observed image, respectively, and $\mathbf{x}$ is the desired high resolution image. $\mathbf{A}_k$ represents the contribution of pixels in $\mathbf{x}$ to the pixels in $\mathbf{y}_k$ due to motion, blurring, and downsampling of the high resolution image. In this paper, a nonparametric motion model and a parametric motion model are assumed [7].

3 Proposed Method

Details on the two processes designed for the image reconstruction are described in this section, along with the termination condition of the iteration. Fig. 1

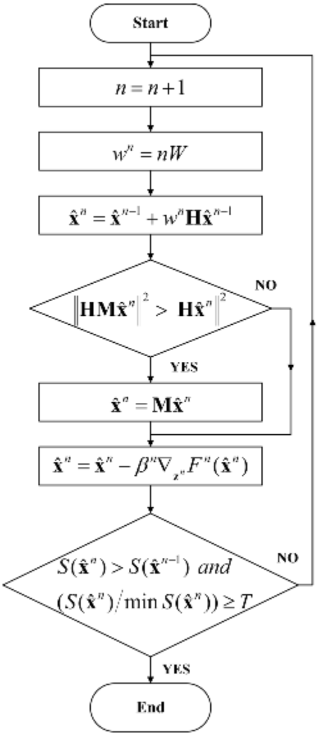

Start
$n = n + 1$
$w^n = nW$
$\hat{\mathbf{x}}^n = \hat{\mathbf{x}}^{n-1} + w^n \mathbf{H}\hat{\mathbf{x}}^{n-1}$
$\|\mathbf{HM}\hat{\mathbf{x}}^n\|^2 > \|\mathbf{H}\hat{\mathbf{x}}^n\|^2$
NO
YES
$\hat{\mathbf{x}}^n = \mathbf{M}\hat{\mathbf{x}}^n$
$\hat{\mathbf{x}}^n = \hat{\mathbf{x}}^n - \beta^n \nabla_{z^n} F^n(\hat{\mathbf{x}}^n)$
$S(\hat{\mathbf{x}}^n) > S(\hat{\mathbf{x}}^{n-1})$ *and*
$(S(\hat{\mathbf{x}}^n)/\min S(\hat{\mathbf{x}}^n)) \geq T$
NO
YES
End

Fig. 1. Flowchart of the proposed iterative algorithm

illustrates the entire iterative process and the rest of the section explains details on functional blocks in the figure.

3.1 Enhancing Edges and Reducing Noise

Edges in the high resolution image are enhanced by introducing a sharpening filter and the process is described as (2).

$$\hat{\mathbf{x}}^n = \hat{\mathbf{x}}^{n-1} + w^n \mathbf{H}\hat{\mathbf{x}}^{n-1} \tag{2}$$

where $\mathbf{H}$ and $n(n \geq 0)$ represent the high pass filter (the second-order derivative in our method) and the iteration order, respectively, and w^n represents the weight parameter of sharpening filter. The degree of edge sharpening increases as the weight parameter , w^n, increases. However, the noise in the image being restored is also amplified in proportion to the degree of edge sharpening.

The noise in the image being restored includes error from the motion estimation and additive zero mean Gaussian noise. On the other hand, it should be noted that the noise in the image is decreased as n increases as well because the error energy for each low resolution image is decreased in the process explained below. In our implementation, we control the weight parameter as shown in (3) by taking into account the error energy of the image being restored. That is, the degree of edge sharpening gradually increases as the iteration number increases.

$$w^n = nW \tag{3}$$

where W represents the basis weight parameter which controls increasing rate of the weight parameter. The basis weight parameter should be small enough to adopt high pass filter stably. The weight control strategy prevents from drastic increase of noise as the iteration continues. Then the edge enhancement process defined in (2) satisfies the inequality shown in (4). That is, the high frequency energy of image being restored is increased by edge enhancement process.

$$\|\mathbf{H}\hat{\mathbf{x}}^{n-1}\|^2 < \|\mathbf{H}(\hat{\mathbf{x}}^{n-1} + w^n\mathbf{H}\hat{\mathbf{x}}^{n-1})\|^2 \tag{4}$$

Even our effort to minimize the undesirable impact, which may occur due to noise during the edge enhancement, has been considered as in (3), noise components in non-edge areas could be amplified during the edge enhancement process. To alleviate this problem we employ a median filter which removes noise effectively while preserving discontinuities. The filter is conditionally activated only when the inequality shown in (5) is satisfied for the high resolution image under restoration.

$$\|\mathbf{HM}\hat{\mathbf{x}}^n\|^2 > \|\mathbf{H}\hat{\mathbf{x}}^n\|^2 \tag{5}$$

where $\mathbf{M}$ represents the median filter. That is, the filtering is applied with conservative manner so that edges in the high resolution image under reconstruction prevent from being smoothen.

3.2 Fidelity and Smoothness Constraints

Utilization of a priori knowledge on $\mathbf{x}$ in (1) plays an important role in image restoration. In contrast to the conventional CLS algorithms where the high frequency energy of the image increases as the iteration number increases, the high frequency energy of the image being restored does not necessarily increases in the constraints satisfaction process because the process of edge enhancement and noise reduction is embedded into the constraints satisfaction process. In short, the result of the edge enhancement process is effectively managed by the application of the smoothness constraint.

As shown in (3), the degree of edge enhancement is proportional to n. Therefore, when n is small the high frequency energy of the restored image increases in the constraints satisfaction process since the increment of the high frequency energy of the image is small in the process of edge enhancement. On the other hand, as n gradually increases, the high frequency energy of the restored image decreases and the edges are smoothed in the constraints satisfaction process since the increment of the high frequency energy of high resolution image is getting larger along with edge enhancement. In each iteration, the constraints satisfaction process is executed based on the following objective function shown in (6).

$$F^n(\mathbf{z}^n) = \sum_k \lambda_k \|\mathbf{y}_k - \mathbf{A}_k\mathbf{z}^n\|^2 + \|\mathbf{C}\mathbf{z}^n\|^2 \tag{6}$$

where λ_k represents the regularization parameter which controls the tradeoff between the fidelity and the smoothness constraint, and $\mathbf{C}$ is the 2-D Laplacian operator [8].

In the conventional CLS algorithm, the objective function is defined once and the result of the previous iterative procedure is used iteratively while minimization of the objective function is being performed. In the proposed method, however, we find a restored image $\mathbf{z}^n$, which satisfies the rule shown in (7), in n^{th} iteration.

$$F^n(\mathbf{z}^n) < F^n(\mathbf{z}^n_{initial}) \tag{7}$$

where $\mathbf{z}^n_{initial}$ represents the initial estimate of $\mathbf{z}^n$ and it is obtained from the process of the edge enhancement and the noise reduction. $\mathbf{z}^n$, which satisfies (7), becomes $\hat{\mathbf{x}}^n$ in our previous convention, and the image being restored is updated using (8).

$$\hat{\mathbf{x}}^n = \hat{\mathbf{x}}^n - \beta^n \nabla_{\mathbf{z}^n} F^n(\hat{\mathbf{x}}^n) \tag{8}$$

where β^n and $\nabla_{\mathbf{z}^n} F^n(\hat{\mathbf{x}}^n)$ are the stepsize and the gradient of the objective function, respectively.

3.3 Termination of Iteration

We use (9) for determining the point where the iteration ends. To reflect our efforts for preserving edges and minimizing noise in the restored image, the equation includes the error energy for each low resolution image and the high frequency energy of the restored image.

$$S(\hat{\mathbf{x}}^n) = \sum_k \|\mathbf{y}_k - \mathbf{A}_k\hat{\mathbf{x}}^n\|^2/p + \|\mathbf{C}\hat{\mathbf{x}}^n\|^2 \tag{9}$$

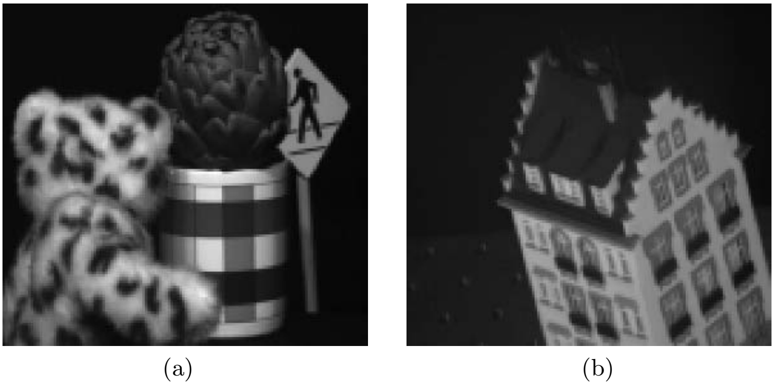
In (9), $S(\hat{\mathbf{x}}^n)$ is decreased when n is small, and then is increased as n increases since the increment in the high frequency energy of the restored image is proportional to n in the process of edge enhancement. That is, when n is large, the high frequency energy of the restored image increased by (2) is larger than that decreased by (8). At the same time, $\|\mathbf{y}_k - \mathbf{A}_k\hat{\mathbf{x}}^n\|^2$ is decreased when n is small and is increased as n increases. Hence, the iteration is terminated when the condition in (10) is satisfied.

$$S(\hat{\mathbf{x}}^n) > S(\hat{\mathbf{x}}^{n-1}) \text{ and } \frac{S(\hat{\mathbf{x}}^n)}{\min S(\hat{\mathbf{x}})} \geq T \tag{10}$$

where $\min S(\hat{\mathbf{x}})$ represents the minimum value of $S(\hat{\mathbf{x}})$ and T is a predefined value.

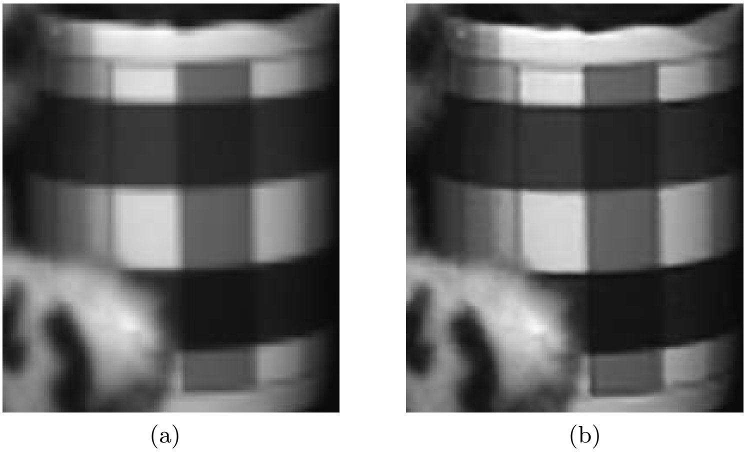
4 Experimental Results

In the experiment, we use `artichoke` and `hotel` video sequences and chose $p = 5$ and $1 < T < 1.1$. The high resolution frame is of size 512 × 480. A



(a) (b)

Fig. 2. Low resolution frames: (a) `artichoke` video sequence and (b) `hotel` video sequence



(a) (b)

Fig. 3. Details of reconstruction results for `artichoke` video sequence: (a) bilinear interpolation and (b) the proposed algorithm

4 × 4 uniform support blurring function is used to obtain a sequence of low resolution frames and a nonparametric motion model is assumed. The result of the proposed method is compared with ones from the bilinear interpolation and the adaptive CLS algorithm [4]. Fig. 2 shows the result low resolution frames for the `artichoke` and `hotel` video sequences. In the performance analysis, we conduct subjective visual evaluation as well as quantitative comparison.

The details of reconstruction results from the bilinear interpolation and the proposed algorithm for the `artichoke` video sequence are shown in Fig. 3 for the comparison purpose. As we observe in the figure, the proposed method produces sharper edges in the restored high resolution images.

The comparison of reconstruction results from an adaptive CLS algorithm [4] and the proposed algorithm is shown in Fig. 4 and Fig. 5. It should be noted

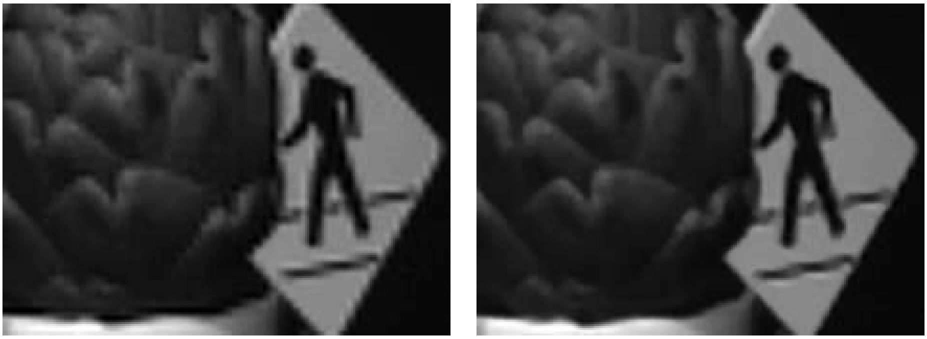

(a) (b)

Fig. 4. Details of reconstruction results for `artichoke` video sequence: (a) adaptive CLS algorithm and (b) the proposed algorithm

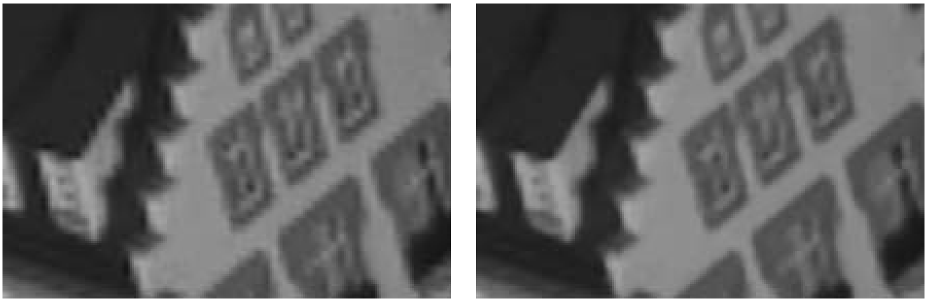

(a) (b)

Fig. 5. Details of reconstruction results for `hotel` video sequence: (a) adaptive CLS algorithm and (b) the proposed algorithm

Table 1. The comparison of average PSNR

	Bilinear interpolation	Adaptive CLS algorithm	Proposed algorithm
`artichoke`	27.43dB	34.43dB	35.67dB
`hotel`	30.18dB	36.58dB	37.37dB

that the proposed algorithm produces high resolution images with sharper edges without noise amplification comparing to the adaptive CLS algorithm. Especially, the staircase effect appears in Fig. 4(a) and Fig. 5(a) is not observed in Fig. 4(b) and Fig. 5(b).

The average PSNRs measured for quantitative evaluation are shown in Table 1. As we can see in the table, the proposed method performs better in terms of PSNR.

To study the sensitivity of the algorithm to the initialization factor, PSNR and the number of iterations versus the basis weight parameter, W, are shown in

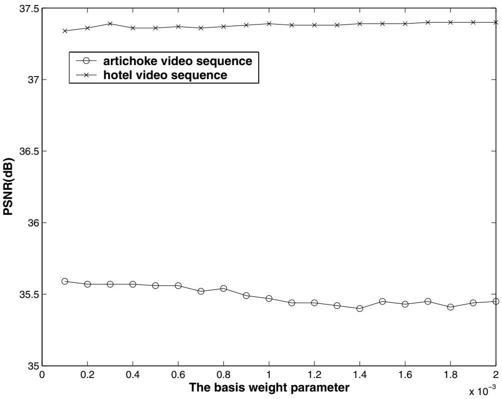


Fig. 6. PSNR versus the basis weight parameter

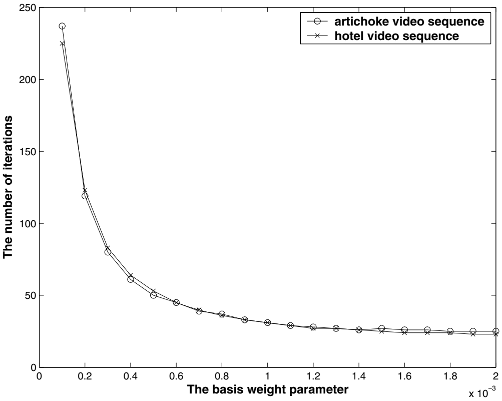


Fig. 7. The number of iterations versus the basis weight parameter

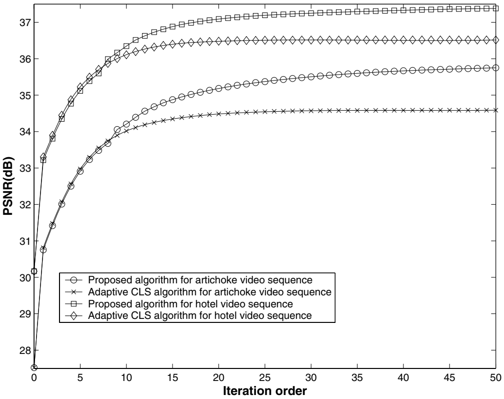
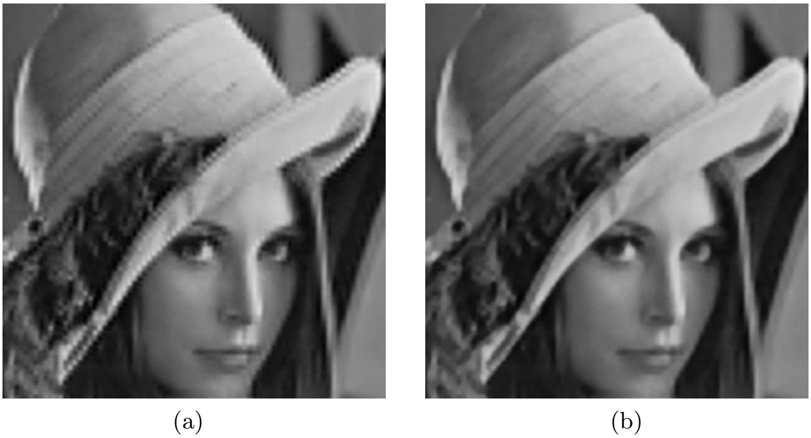
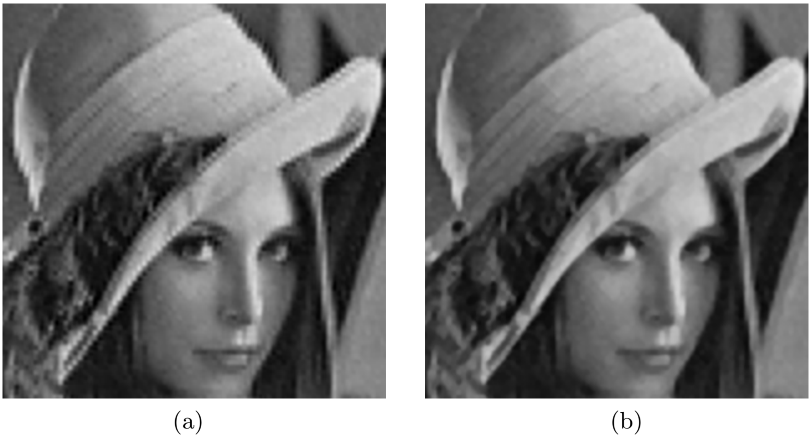


Fig. 8. PSNR versus iteration order



(a) (b)

Fig. 9. Details of reconstruction results for `lena` video sequence: (a) adaptive CLS algorithm and (b) the proposed algorithm



(a) (b)

Fig. 10. Details of reconstruction results for `lena` video sequence with Gaussian noise ($\sigma = 3$): (a) adaptive CLS algorithm and (b) the proposed algorithm

Fig. 6 and Fig. 7, respectively. As we can see in Fig. 6, the proposed method is not much sensitive to the basis weight parameter in terms of PSNR for both of the image sequences. On the other hand, as Fig. 7 indicates, the number of iterations is directly influenced by the basis weight parameter since the termination point is depending on the initialization factor.

Fig. 8 shows how PSNR varies as the iteration order increases. First of all, from the figure it should be noted that the proposed method results in almost the same PSNR to the CLS algorithm with quite comparable iteration number until the point where the CLS algorithm converges. However, the figure indicates that the proposed method is able to produce high resolution images with better PSNR by considering a few number of additional iteration.

Fig. 9 and Fig. 10 show reconstruction results for lena and noisy lena video sequence, respectively. The high resolution frame is of size 512 × 512. To obtain a noisy sequence of low resolution frames, a 4 × 4 uniform support blurring function is employed and Gaussian noise with standard deviation 3 is added. A parametric motion model is assumed. As we can see in Fig. 9(b) and Fig. 10(b), the edges are almost completely preserved with presence of noise.

5 Conclusion

In this paper, we propose a new iterative approach for high resolution image reconstruction which is preserving edges but reducing noise. The efforts for preserving edges and reducing noise have been effectively embedded into a framework which is considering the fidelity and the smoothness constraints satisfaction process. Experimental results presented in section 4 prove that the proposed iterative algorithm performs better than the conventional ones including a bilinear interpolation and an adaptive CLS reconstruction algorithm, in terms of not only PSNR but also perceived visual quality. In addition, it turns out that the proposed algorithm is less sensitive to the initialization factor as well.

References

1. Elad, M., Feuer, A.: Restoration of a single superresolution image from several blurred, noisy and undersampled measured images. IEEE Transactions on Image Processing **6** (1997) 1646–1658
2. Altunbasak, Y., Patti, A.J., Mersereau, R.M.: Super-resolution still and video reconstruction from mpeg-coded video. IEEE Transactions on Circuits and Systems for Video Technology **12** (2002) 217–226
3. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: A technical overview. IEEE Signal Processing Magazine **20** (2003) 21–36
4. Lee, E.S., Kang, M.G.: Regularized adaptive high-resolution image reconstruction considering inaccurate subpixel registration. IEEE Transactions on Image Processing **12** (2003) 826–837
5. Schultz, R.R., Stevenson, R.L.: Extraction of high-resolution frames from video sequences. IEEE Transactions on Image Processing **5** (1996) 996–1011
6. You, Y.L., Kaveh, M.: A regularization approach to joint blur identification and image restoration. IEEE Transactions on Image Processing **5** (1996) 416–428
7. Schultz, R.R., Meng, L., Stevenson, R.L.: Subpixel motion estimation for super-resolution image sequence enhancement. Journal of Visual Communication and Image Representation **9** (1998) 38–50
8. Hardie, R.C., Barnard, K.J., Bognar, J.G., Armstrong, E.E., Watson, E.A.: High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system. Optical Engineering **37** (1998) 247–260

Fast Binary Dilation/Erosion Algorithm Using Kernel Subdivision

Ajay Narayanan

Imaging Technologies Lab, GE Global Research, Bangalore, India
ajay.narayanan@ge.com

Abstract. Numerous algorithms have been proposed in the literature to speed up dilation/erosion operations. The motivation has been to reduce computational complexity by exploiting the structuring element and the image object properties. This paper presents a new algorithm for binary morphological dilation and erosion called the Kernel Sub-Division algorithm and discusses its implementation in the two dimensional case. It decomposes the n-dimensional structuring element, into several subsets and operates on the object contours in the image. The image characteristics are exploited by subdividing the object contours into bins while performing contour processing. The elegance of the algorithm lies in its retaining the correspondence to the output of the classical implementation with massive speed gain. The results of the algorithm on a statistically significant test set of images, showed that it performed five times better than the classical implementation for a 3x3 kernel. It also demonstrated a marginal rise in execution time with increasing size of the kernel.

1 Introduction

Morphology has a strong mathematical basis in Set theory [1]. The dilation of a set of points X by a structuring element B is defined [2] as follows.

$$X \oplus B = \{x + b | ((x \in X) \wedge (b \in B))\} \tag{1}$$

Erosion is the dual of dilation. A direct implementation of the definition above leads to a computational cost of complexity $o(n^2 \cdot A)$ for an image of size n x n and a kernel of size d x d with area A. Elementary decomposition of structuring elements is used to reduce the complexity of dilation in [2]. Further speed up is achieved in literature [3][4][5] by reducing computational complexity at the cost of sacrificing correspondence to the classical implementation output. Considering only the edge pixels of the set X the exact correspondence to the output of the classical algorithm can be maintained [2].

$$X \oplus B = X \cup (\delta(X) \oplus B) \tag{2}$$

where $\delta(X)$ is the edge of X [2]. If,

$$\eta(\delta(X)) = l \tag{3}$$

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 335–342, 2006.

the computational complexity then reduces to

$$o(l \cdot d^2) \tag{4}$$

The motivation of this work has been to develop a fast algorithm while matching the output of the classical definition of the dilation/erosion operations. The elegance of the Kernel sub-division (KSD) algorithm is that it achieves reduction in computational complexity while maintaining the exact output as that of the classical definition implementation for both dilation and erosion. It also sets up a framework to handle anisotropic dilation and extension into 3-dimensional volume dilation. This paves the way for the discussion of the Kernel sub-division (KSD) algorithm. This will be followed by a brief discussion on the 2D implementation of KSD, results and future extensions.

2 Methods

A brief overview of the contour processing method with a full kernel implementation is presented below. A circular kernel of size d x d is chosen for illustrative purposes as in Figure 1.

Fig. 1. (a) A circular kernel, (b) An object in an image

Figure 2 shows the way in which a full-kernel contour processing modifies the objects when dilating them.

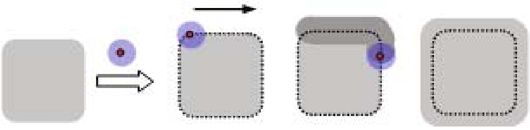


Fig. 2. The process of dilation using full kernel contour processing

In the full kernel contour processing method, the kernel modifies the pixels that are interior to the object. The Kernel subdivision method exploits this computational redundancy by devising a framework in which only the pixels exterior to the object are modified in case of dilation. This would then give a gain factor of two in the processing time. This concept is illustrated in Figure 3.

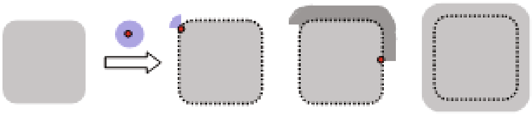


Fig. 3. The process of dilation using kernel subdivision contour processing

The Kernel subdivision method approaches the problem from both the kernel and the image perspective by optimizing both the l and the d^2 term in (4).

2.1 Contour Binning

The KSD algorithm fractionates both the l and the d^2 terms. A representative object in an image is shown in Figure 4.

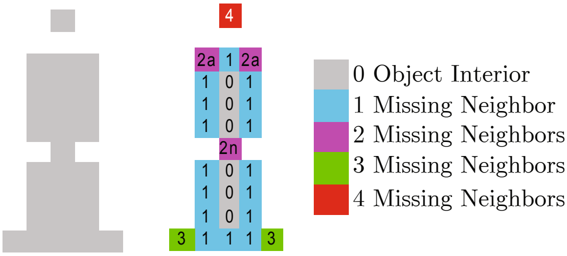


Fig. 4. A representative object to illustrate contour binning

The term l is modified by binning the contour pixels into 5 bins such that,

$$l = l_1 + l_{2adj} + l_{2n-adj} + l_3 + l_4 \tag{5}$$

where:

l_1 = Set of pixels having only one 4-connected neighbor missing. (shown as 1 in the contour map in Figure 4)

l_{2adj} = Set of pixels having only two 4-connected neighbors missing which are in turn perpendicular to each other. (Shown as 2a in the contour map in Figure 4)

l_{2n-adj} = Set of pixels having only two 4-connected neighbors missing which in turn are not in set l_{2adj} (shown as $2n$ in the contour map in Figure 4).

l_3 = Set of pixels having only three 4-connected neighbors missing (shown as 3 in the contour map in Figure 4).

l_4 = Set of pixels having only four 4-connected neighbor missing (shown as 4 in the contour map in Figure 4).

2.2 Kernel Subdivision

The term d^2 is modified by sub-dividing the kernel into a set of oriented 2D subkernels. These subkernels are so chosen that each one applies to a particular bin of l. These are precomputed and stored in a lookup table. A circular kernel is chosen for illustration of kernel subdivisions in Figure 5.

The kernel subdivision used depends on the contour bin into which the image contour pixel falls. This stems from the observation that the actual region to be di-

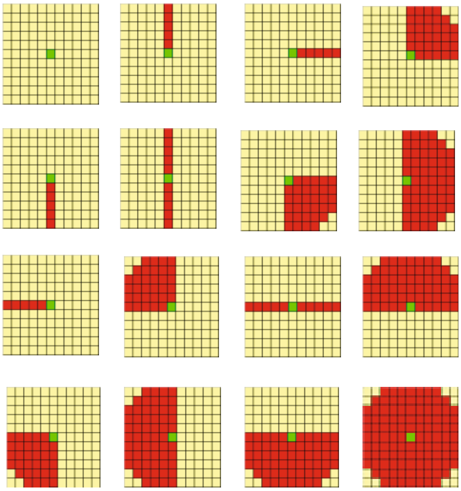


Fig. 5. The 16 linear combinatorial sub-divided kernels of a 2D Circular kernel for d=11 [read 0-15 left to right, top to bottom]

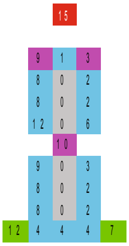


Fig. 6. The numbers in the pixels indicate the subkernel to use from the lookup in Figure 5 for a contour bin

lated depends upon the neighborhood conditions of a contour pixel as illustrated in Figure 6.

The following is the mapping relationship of subdivided kernels (right side) and the contour bins they link up to (left side).

$$l_1 \leftrightarrow \{1, 2, 4, 8\}$$

$$l_{2adj} = \{3, 6, 9, 12\}$$

$$l_{2n-adj} = \{5, 10\}$$

$$l_3 = \{7, 11, 13, 14\}$$

$$l_4 = \{15\}.$$

2.3 Computational Complexity

The computational complexity can now be calculated. We have:

For bin 1: $$complexity = o\left\{l_1 \cdot \left(\frac{(d-1)}{2}\right)\right\}$$

For bin 2adj: $$complexity = o\left\{l_{2adj} \cdot \left(\frac{1}{4} \times A\right)\right\}$$

For bin 2n-adj: $$complexity = o\{l_{2n-adj} \cdot d\}$$

For bin 3: $$complexity = o\left\{l_3 \cdot \left(\frac{1}{2} \times A\right)\right\}$$

For bin 4: $$complexity = o\{l_4 \cdot A\}$$

Where A = area of the kernel foreground = $\pi\left(\frac{d^2}{4}\right)$ ·

Total complexity is the linear sum of the individual bin complexities. It can thus be seen that the total order of complexity is a direct function of the statistical distribution of the contour pixels in the contour bins.

Thus the algorithm tunes it's execution speed to the incoming data by exploiting data redundancy.

3 Implementation

3.1 Run Length Encoding of Kernel Foreground

To prevent revisits to pixels in the image which are under a sub-kernel's background, the kernel foreground is run-length encoded. This means only the pixels under the foreground of a sub-kernel are visited when a kernel is applied over a point. The encoding is pre-computed over the image coordinate extents (horizontal and vertical) and stored in memory.

3.2 Positional Code

To assign a particular sub-divided kernel to a contour bin is a tricky task as it is not a direct one-to-one mapping. For each pixel classified as belonging to bin 1 there are 4 possible choices of sub-kernels depending upon the 2D spatial arrangement of neighbors. This is handled by encoding the neighborhood positions for a contour pixel into a 4-bit positional code that maps it to the corresponding sub-kernel in the memory lookup. This is illustrated in Figure7, where 'c' is the centre pixel and 4-connectiviy is used.

3.3 Experiments

The average time to execute this algorithm on the target hardware platforms (1.4 GHz, 2GB, 1 Pentium IV Processor) is compared against the performance of the conventional algorithm in Table1. Input is a set of 275 images with size 512*512 pixels. Time shown in Table1 is the time taken to process the full set of images. The performance was compared against a VTK filter *vtkImageDilateErode3D*. The fast dilation using kernel subdivision algorithm was also implemented in VTK framework. This ensured a fair imaging pipeline of execution to maintain constancy in comparison due to same underlying framework of data and execution.

Bit representation of Positional Code

Top	Right	Bottom	Left
0/1	0/1	0/1	0/1
2^0	2^1	2^2	2^3

0/1 = absence/presence

2^n = position weights

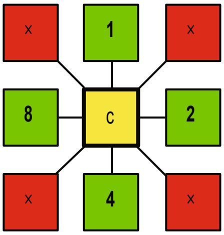


Fig. 7. Position Code

Table 1. Execution time of kernel subdivision algorithm against conventional algorithm

Layer(s)	Time of execution (seconds)		Gain factor
	Kernel subdivision	Conventional	
1	1.203	5.765	4.8
5	1.297	47.562	36.7
10	1.422	157.138	110.5
15	1.578	337.025	213.6
20	1.883	572.426	304.0

4 Results

The Kernel Sub-division algorithm has been applied to a medical CT Angiography dataset. Table1 and Figure 8 show that the KSD algorithm performed better than the standard implementation and from complexity analysis it has been shown

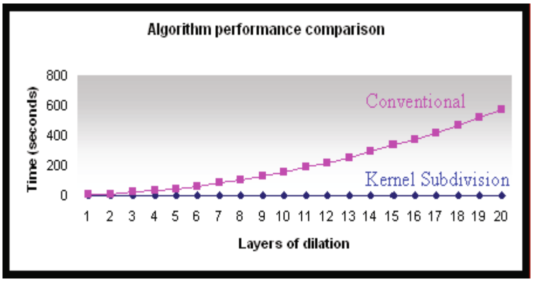


Fig. 8. Execution time comparison against conventional algorithm

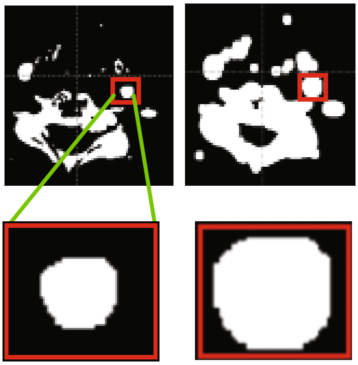


Fig. 9. (a) Image before, (b) After five layers dilation

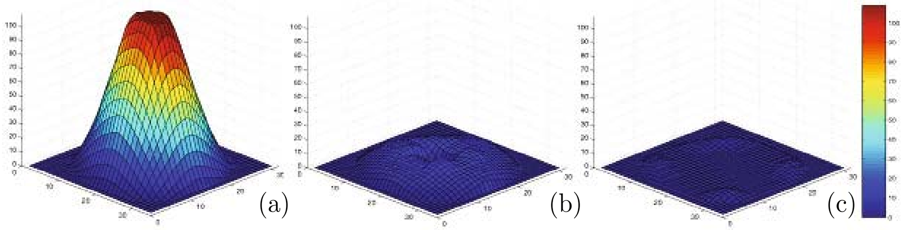


Fig. 10. Pixel revisits/complexity (a) conventional algorithm, (b) full kernel contour algorithm, (c) kernel subdivision

that it can outperform full kernel contour processing techniques. The algorithm was also analyzed for performance against a 'vtk' implementation of full kernel contour based dilation algorithm and was found to have a gain factor of 1.9x. This difference from the predicted gain factor of two is attributed to the small computational overhead of neighborhood positional code encoding, needed to be performed in the Kernel sub-division technique. Figure 9 shows an image that has been dilated by five layers. Figure 10 visually shows the computational complexity for an object in the image (zoomed). The algorithm can be extended to three-dimensional dilation or erosion with minimal modification in the framework. The combinations though reach up to 64 sub-kernels; the rotational invariance of these sub-division kernels can be exploited to have two or more of the positional code pointing to the same sub-division kernel in the lookup.

5 Conclusions

The kernel subdivision algorithm performs twice as fast than the full kernel contour-processing method at all layers. The algorithm is five times faster than the classical algorithm at one layer and upto one hundred times faster at 10 layers. It utilizes image information to speed up its performance. Though the actual computational gain may vary for different images, the algorithm performs well on images with statistically significant distribution of contours in various bins. The work has been extended to do binary mask controlled gray level image dilation with an ability to handle anisotropic kernels. Further extension of the work to three-dimensional dilation/ erosion is in progress.

References

1. Desikachari Nadadur and Robert M. Haralick, Fellow, IEEE, "Recursive Binary Dilation and Erosion Using Digital Line Structuring Elements in Arbitrary Orientations." IEEE Transactions on Image Processing, Vol. 9, No. 5, MAY 2000.
2. Cuisenaire, O. (Universite Catholique de Louvain); Macq, B. "Fast Euclidean morphological operators using local distance transformation by propagation, and applications", IEEE Conference Publication, v 2, n 465, p 856-860, 1999.
3. P. Soille, E. J. Breen, and R. Jones, "Recursive implementation of erosions and dilations along discrete lines at arbitrary angles," IEEE Trans. Pattern Anal. Machine Intell., vol. 18, pp. 562–567, May 1996.
4. I. Ragnemalm, "Fast erosion and dilation by contour processing and thresholding of distance maps," Pattern Recognit. Let., vol. 13, pp. 161–166, 1985.
5. Parker, J.R. (Univ of Calgary), "System for fast erosion and dilation of bi-level images", Journal of Scientific Computing v5, n 3, p 187-198, Sep, 1990.

Fast Global Motion Estimation Via Iterative Least-Square Method

Jia Wang, Haifeng Wang, Qingshan Liu, and Hanqing Lu

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100080, China
{wangjia, hfwang, qsliu, luhq}@nlpr.ia.ac.cn

Abstract. This paper presents a fast algorithm for global motion estimation based on Iterative Least- Square Estimation (ILSE) technique. Compared with the traditional framework, three improvements were made to accelerate the computation progress. First, a new 3-parameter linear model, together with its solution using modified ILSE method, is proposed to describe and estimate global motion, which is simple and reasonable. Second, a pre-analysis method, Gradient Thresholding (GT) method, is introduced to pre-analyze the image macro-blocks before global motion estimation using their gradient information, which reduce the computational cost by reducing the amount of involved blocks. Lastly, Successive Elimination Algorithm (SEA), which is used to calculate motion field, is improved by a new presented matching criterion considering both the gradient information and the intensity information. The presented method has been tested on a variety of image sequences, and experimental results illustrate its promising performance.

1 Introduction

Motion estimation is one of the research topics having attracted many research activities in the video compression and coding community [1]. In video sequence, motion always comes from the movement of camera, movement of objects in the scene, or movement of both. The former is often referred as global motion and the latter as local motion. Separating these two classes of motion, which is the main job for Global Motion Estimation (GME), is significant for video coding, video indexing, video object segmentation, and many other applications.

Iterative Least Square Estimation (ILSE) technique is a commonly used method for global motion estimation. Recently, Rath and Makur [2] proposed a four- parameter model to calculate global motion parameters using ILSE. Their method consists of two steps: First, an initial motion field is calculated using Block Matching Algorithm (BMA) [3] considering all of the blocks in a frame. Because some calculated motion vectors will be inaccurate for the blocks, in the second step, ILSE technique is used to gradually eliminate the influence of these blocks and finally extract accurate global motion parameters.

A problem of Rath and Makur's framework is high computational cost, which mainly comes from the motion field estimation using BMA. Recently, Sorwar

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 343–352, 2006.

etc.[4] proposed a Distance Dependent Thresholding (DTS) method for fast block matching. Besides, in their work, only the blocks near the edge of image are concerned in the global motion estimation, which reduce the computational cost to a great extent. Yet, its accuracy highly depends on the amount of concerned blocks.

In this paper, we propose a fast algorithm for global motion estimation based on a 3-parameter linear model and ILSE technique. The new model is deducted by simplifying Rath and Makur's 4-parameter linear model. And the three parameters are estimated using a modified ILSE method. On the other hand, a pre-analysis method, Gradient Thresholding (GT) method, is used to reduce the computation cost, in which all of the blocks are pre-analyzed by their gradient information. The aim is to find those treacherous blocks that are more likely to produce inaccurate motion estimations, and exclude them from the following process. Furthermore, Successive Elimination Algorithm (SEA), which is used to calculate motion field, is improved by a new presented matching criterion considering both the gradient information and the luminance information.

2 Global Motion Estimation

2.1 Three-Parameter Global Motion Model

In this section, 3-parameter model is proposed by simplifying Rath and Makur's 4-parameter model. In [2], Rath and Makur have used two parameters to describe the camera zoom, corresponding to the x-axis and y-axis separately. Yet actually, the zoom factor along x-axis and y-axis should be identical. So, in our work, only one parameter is remained to denote camera zoom factor. The motion model can be expressed as

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = a_1 \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} \tag{1}$$

where

$$a_1 = z_{xy},\ a_2 = f_1(p_x, z_{xy}),\ a_3 = f_2(p_y, z_{xy}) \tag{2}$$

z_{xy} is the zoom factors and (p_x, p_y) is the pan vector.

The motion parameters are calculated using ILSE technique, which involves two steps. First, the frame image is segmented into several $m \times n$ blocks, and block-matching algorithm (BMA) is performed to estimate the motion vector for each block. Second, ILSE technique is used to compute global motion parameters from the estimated motion field constructed by the blocks and their motion vectors.

Let there be N blocks in a video frame, and assume that the motion vector of a block is the motion vector of the central pixel of that block. Let (v_x^k, v_y^k) be the estimated motion vector by BMA, of the block $k(k = 0, 1, \cdots, N-1)$, whose central pixel's coordinates are (s_x^k, s_y^k) with respect to the center of the frame. Then, the global motion model can be written as:

$$\begin{bmatrix} v_x^k \\ v_y^k \end{bmatrix} = a_1 \begin{bmatrix} s_x^k \\ s_y^k \end{bmatrix} + \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} \tag{3}$$

According to the ILSE algorithm, the optimal values for camera parameters (a_1, a_2, a_3) are estimated using the following criteria:

$$(a_1, a_2, a_3) = \arg\min \sum_{k=0}^{N-1} [(v_x^k - a_1 s_x^k - a_2)^2 + (v_y^k - a_1 s_y^k - a_3)^2] \tag{4}$$

Differentiating (4) with respect to the parameters and setting the derivatives to zero, we get

$$\begin{bmatrix} \sum(s_x^k)^2 + \sum(s_y^k)^2 & \sum s_x^k & \sum s_y^k \\ \sum s_x^k & N & 0 \\ \sum s_y^k & 0 & N \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \sum v_x^k s_x^k + \sum v_y^k s_y^k \\ \sum v_x^k \\ \sum v_y^k \end{bmatrix} \tag{5}$$

Then, the following solution can be achieved as:

$$a_1 = \frac{N\Psi_1 - \Psi_2}{N\Psi_3 - \Psi_4} \tag{6}$$

$$a_2 = \Psi_3 \sum v_x^k - \Psi_1 \sum s_x^k + \frac{\Psi_5}{N} \sum s_y^k \tag{7}$$

$$a_3 = \Psi_3 \sum v_y^k - \Psi_1 \sum s_y^k - \frac{\Psi_5}{N} \sum s_x^k \tag{8}$$

where

$$\Psi_1 = \sum v_x^k s_x^k + \sum v_y^k s_y^k, \Psi_2 = \sum v_x^k \sum s_x^k + \sum v_y^k \sum s_y^k, \tag{9}$$

$$\Psi_3 = \sum (s_x^k)^2 + \sum (s_y^k)^2, \Psi_4 = (\sum s_x^k)^2 + (\sum s_y^k)^2, \tag{10}$$

$$\Psi_5 = \sum v_y^k \sum s_x^k + \sum v_x^k \sum s_y^k. \tag{11}$$

To avoid the influence of the blocks with inaccurately estimated motion, the above procedure is evaluated iteratively, and each iteration eliminates blocks whose motion vectors (estimated by BMA) do not match with the current global motion fields. Matching means that a motion vector lies within a threshold distance from the corresponding global motion field. So, using ILSE method, the influence of those blocks with inaccurate motion estimations will be gradually removed, and after several iterations, the estimated parameters will converge to the final results.

2.2 Modified ILSE Method

Although ILSE method can avoid estimation bias caused by inaccurate motion vectors, it is weak to dispose of another problem which may also cause bias in global parameter estimation: the existing of local motions. The reason is: inaccurate motion vectors are always disordered and can be discarded easily, while the local motions are not. Such problem becomes even worse when the objects are large in the scene.

Based on the prior knowledge that moving objects generally exist towards the centre of frame, the weights of blocks for estimation are adjusted according to their distances with respect to the centre of frame. Then equation (4) is modified as

$$f(a_1, a_2, a_3) = \sum_{k=0}^{N-1} [(v_x^k - a_1 s_x^k - a_2) s_x^k]^2 + [(v_y^k - a_1 s_y^k - a_3) s_y^k]^2 \tag{12}$$

based on what, (a_1, a_2, a_3) can be computed by

$$a_1 = \frac{(\sum v_x s_x^3 + \sum v_y s_y^3) \sum s_x^2 \sum s_y^2 - \sum v_x s_x^2 \sum s_x^3 - \sum v_y s_y^2 \sum s_y^3}{(\sum s_x^2 + \sum s_y^2) \sum s_x^2 \sum s_x^4 - (\sum s_x^3)^2 - (\sum s_y^3)^2} \tag{13}$$

$$a_2 = \frac{\sum v_x s_x^2 - a_1 \sum s_x^3}{\sum s_x^2} \tag{14}$$

$$a_3 = \frac{\sum v_y s_y^2 - a_1 \sum s_y^3}{\sum s_y^2} \tag{15}$$

In the above formulas, the subscript k is omitted for simplification purpose.

Based on the above criterion (12) and related formulas, blocks near the outer part of the image will produce a more significant effect on the estimation of global motion parameters. And those near the center of the image, which are more likely to be part of a moving object, will have less significance.

3 Gradient Thresholding

In [2], Rath and Makur considered all the rows and columns of macroblocks in a frame to estimate the global motion parameters. Since many blocks will finally be excluded for their inaccurately estimated motion vectors, it's not necessary to involve all the blocks into the task. In this section, Gradient Thresholding (GT) method is proposed to preprocess the blocks considering gradient information before the motion estimation. The aim is to identify those treacherous blocks that are more likely to produce inaccurate motion estimations.

For that purpose, the image should first be processed to make the Gradient Map. Many methods can be used for this task, such as the Sobel Operator, Roberts Operator [5], etc. Using gradient map, the gradient information of each block will be checked as follows.

For a block B with $m \times n$ pixels, let $p_{i,j}$ be the pixel with coordinates (i, j) in the block and gradient $G(p_{i,j})$. The gradient value of block B is defined as

$$G_B = \sum_{i=1}^{m} \sum_{j=1}^{n} [G(p_{i,j})/n^2] \tag{16}$$

which is the mean gradient of all the pixels in B.

For all the blocks in the frame, we define a threshold θ to classify them as:

A block B will be treated as a treacherous block, when its gradient value $G_B < \theta$.

As soon as a block is decided to treacherous, it will not be considered in the following process by BMA and ILSE. An example of gradient thresholding is shown in Fig. 1.

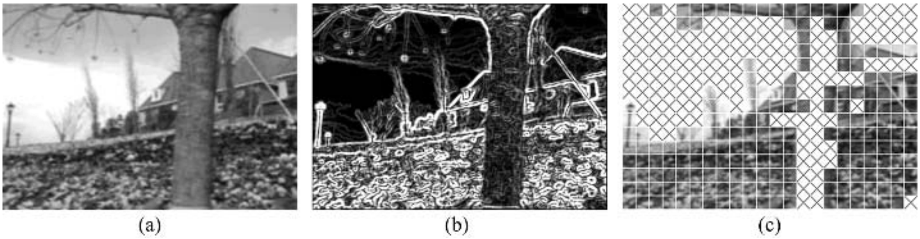


Fig. 1. Gradient thresholding: (a) the test image is taken from *Flower Garden* sequence; (b) shows the computed gradient map; in (c), the treacherous blocks recognized by GT method are marked by ×. The threshold is $\theta = 70$.

As shown in Fig. 1.(c), there are totally 330 blocks whose size are 16×16. After gradient thresholding, 145 blocks (about 44% to 330) is marked as treacherous and will not be considered in the following operation.

To automatically select threshold, considering both the computational cost and estimation accuracy, the relationship between amount of treacherous blocks and motion estimation accuracy has been studied (as shown in Fig. 2). We found that, in many cases, if there were less than 30% blocks remaining (viz. more than 70% blocks are treacherous), the estimated parameters would lose their accuracy.

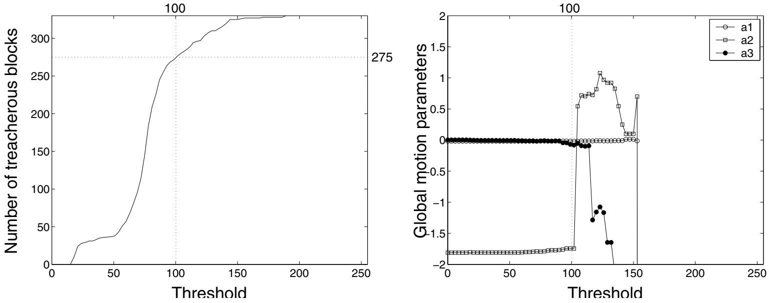


Fig. 2. Threshold analysis: The left graph illustrates that the amount of treacherous blocks increases along with the selected threshold θ. When θ goes higher, more treacherous blocks will be discarded and the computational cost will reduce. But on the other hand, θ can't go higher without limit. From the right graph, it can be seen that when θ is higher than 100, the estimated parameters lost their accuracy.

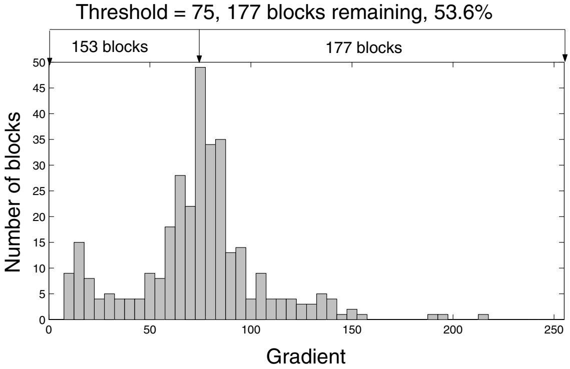


Fig. 3. Gradient histogram is used to automatically choose the threshold θ, which allows 50% of the blocks to pass through the GT process

Considering the diversity of different image sequences, in our work, we choose such threshold that about 50% of the blocks can pass through the GT process. Gradient histogram can used for the threshold selection task, as shown in Fig. 3.

4 Motion Field Calculation

In this section, motion field is calculated using the Successive Elimination Algorithm (SEA) [6][7][8][9]. SEA is a fast method for block matching, which introduces SAD (Sum of Absolute Difference) or MSE (Mean Squared Error) as matching criterion. Before calculating the criterion, SEA first uses its lower bound to check the search positions. If a position can't pass the bound, it will be excluded directly and won't be considered in the criterion calculation. Because the calculation of criterion's lower bound is easier than that of itself, much time is saved by SEA using the lower bound.

The commonly used criteria for SEA are SAD and MSE, which only considering intensity information. In this section, we introduce an extra criterion for SEA, named SAG (Sum of Absolute Gradient difference), and SAG is combined with SAD to improve the performance of SEA.

Let I_t and I_{t+1} be the consecutive frames for motion estimation. In frame I_t, a pixel with coordinates $\mathbf{x} = (x, y)^T$ has gradient $g(\mathbf{x}, t) : 0 \leq g(\mathbf{x}, t) \leq 2^8 - 1$. On the other hand, I_{t+1} is segmented into J blocks G_j, each including N pixels. Then the gradients of the pixels can be described by an N-dimension vector

$$\mathbf{g}_{t+1}^{j} = [g_1, g_2, \cdots, g_N]^T \tag{17}$$

During block matching using SAG, each block in I_{t+1} will be compared with K positions in I_t, using their N-dimension gradient vector $\mathbf{g}_{t+1}^{j}$ and $\mathbf{g}_{t,k}^{j}$, to find the best match.

Let $\|\mathbf{a}\|_p = \sqrt[p]{|a_1|^p + |a_2|^p + \cdots + |a_N|^p}$ be the p-norm of N-dimension vector $\mathbf{a}$. The matching criterion based on SAG is defined as

$$\phi_{SAG}(k) := \|\mathbf{g}_{t+1} - \mathbf{g}_{t,k}\|_1 \tag{18}$$

The search for the SAG-optimal position k_{SAG} in I_t can be expressed as

$$k_{SAG} = \arg\min_k \phi_{SAG}(k) \tag{19}$$

SEA uses lower bounds to pre-analyze positions. Let $\mathbf{u}_N = [1, 1, \cdots, 1]^\top$ denote the length-N column vector with all elements equal to 1. Based on triangular inequality, there is

$$|(\mathbf{u}_N)^\top \mathbf{g}| \leq \|\mathbf{g}\|_1 \tag{20}$$

$\mathbf{a}$ is chosen as $\mathbf{g}_{t+1} - \mathbf{g}_{t,k}$. Then equation (20) can be written as

$$|\|\mathbf{g}_{t+1}\|_1 - \|\mathbf{g}_{t,k}\|_1| \leq \|\mathbf{g}_{t+1} - \mathbf{g}_{t,k}\|_1 \tag{21}$$

Together with (18), there is

$$|\|\mathbf{g}_{t+1}\|_1 - \|\mathbf{g}_{t,k}\|_1| \leq \phi_{SAG}(k) \tag{22}$$

Let

$$\Phi_{SAG}(k) := |\|\mathbf{g}_{t+1}\|_1 - \|\mathbf{g}_{t,k}\|_1| \tag{23}$$

$\Phi_{SAG}(k)$ is the required lower bound for SAG at search position k.

Using SAG, SEA matches the blocks as follows:

Step 1. There are K positions in I_t to be compared with G_j of I_{t+1}. Let $k = 1, 2, \cdots, K$;

Step 2. When $k = 1$, $\hat{\phi}_1 = \phi_{SAG}(1) = \|\mathbf{g}_{t+1} - \mathbf{g}_{t,1}\|_1$ is calculated for G_j as the initial ϕ_{SAG};

Step 3. For $k = 2, 3, \cdots, K$, a smaller ϕ_{SAG} can only be found in the search position k if $\Phi_{SAG}(k) < \hat{\phi}_{k-1}$ is satisfied. In other words, a position k with $\Phi_{SAG}(k) \geq \hat{\phi}_{k-1}$ will be excluded directly without calculating $\phi_{SAG}(k)$. $\hat{\phi}_k$ is refreshed as follows

$$\hat{\phi}_k = \begin{cases} \hat{\phi}_{k-1} & if\ \Phi_{SAG}(k) \geq \hat{\phi}_{k-1} \\ \min(\phi_{SAG}(k), \hat{\phi}_{k-1}) & if\ \Phi_{SAG}(k) < \hat{\phi}_{k-1} \end{cases} \tag{24}$$

Step 4. If all K search positions have been examined, $\hat{\phi}_k$ corresponds to the smallest ϕ_{SAG} for G_j, and the SAG-optimal position k is found.

Using SAG, SEA needs to calculate the 1-norm $\|\mathbf{g}\|_1$ of every block first. Since gradient information is already available during the gradient thresholding stage, $\|\mathbf{g}\|_1$ can be calculated by the same fast method as used for SAD [7][10].

Combining SAG and SAD, tighter bounds can be deduced for SEA. Because more information is considered to pre-analyze blocks before calculating matching criterion, more search positions are excluded, which lead to a faster matching of blocks.

5 Experimental Results

The proposed method was tested on a variety of image sequences, some of which are shown in this section. Note that in our experiments, Peak Signal-to-Noise Ratio (PSNR) is used to measure the estimation accuracy of different method.

Table 1 shows the statistical simulation results achieved from several testing sequences. It can be observed that SEA using SAD&SAG has a similar perfor-

Table 1. Statistical performance comparison of SEA

Sequences (Format)	Comparison	SEA (SAD)	SEA (MSE)	SEA (SAG)	SEA (SAD&SAG)
Mobile-calendar	PSNR(dB)	23.98	23.92	23.27	**23.85**
CIF 352 × 288	Calc. time(ms/frame)	210	261	237	**136**
Foreman	PSNR(dB)	30.92	30.97	29.39	**30.07**
CIF 352 × 288	Calc. time(ms/frame)	206	226	202	**93**
Table-tennis	PSNR(dB)	28.31	28.37	26.95	**28.02**
SIF 352 × 240	Calc. time(ms/frame)	224	276	263	**113**
Hallman	PSNR(dB)	28.12	27.99	27.70	**28.06**
SIF 352 × 240	Calc. time(ms/frame)	196	221	216	**91**
Hall-monitor	PSNR(dB)	34.21	34.38	33.66	**33.87**
QCIF 176 × 144	Calc. time(ms/frame)	30	34	25	**16**
Coastguard	PSNR(dB)	31.90	31.78	29.84	**31.68**
QCIF 176 × 144	Calc. time(ms/frame)	42	52	49	**24**

Table 2. Statistical performance comparison of GME

Sequences (Format)	Comparison	S&M (4 parameters)	Proposed method (3 parameters)
Mobile-calendar	PSNR(dB)	19.86	**19.89**
CIF 352 × 288	Calc. time(ms/frame)	328	**99**
Foreman	PSNR(dB)	25.55	**25.62**
CIF 352 × 288	Calc. time(ms/frame)	390	**71**
Table-tennis	PSNR(dB)	22.08	**23.61**
SIF 352 × 240	Calc. time(ms/frame)	295	**99**
Hallman	PSNR(dB)	26.03	**26.52**
SIF 352 × 240	Calc. time(ms/frame)	207	**55**
Hall-monitor	PSNR(dB)	33.45	**33.55**
QCIF 176 × 144	Calc. time(ms/frame)	138	**18**
Coastguard	PSNR(dB)	25.13	**25.15**
QCIF 176 × 144	Calc. time(ms/frame)	173	**17**

Table 3. Runtime of proposed method

Frame number	Gradient thre.	1-norm calc.	SEA&ILSE	Total
Mobile-calendar(299)	1170ms(3.9%)	6891ms(23.3%)	21584ms(72.8%)	29645ms
Foreman(299)	1219ms(5.8%)	7094ms(33.5%)	12843ms(60.7%)	21156ms
Table-tennis(149)	393ms(2.7%)	2766ms(18.7%)	11608ms(78.6%)	14767ms
Hallman(88)	234ms(4.8%)	1669ms(34.3%)	2965ms(60.9%)	4868ms
Hall-monitor(329)	375ms(6.2%)	1785ms(29.5%)	3892ms(64.3%)	6052ms
Coastguard(299)	361ms(7.0%)	1608ms(31.4%)	3154ms(61.6%)	5123ms
Average	4.6%	26.7%	68.7%	100%

mance compared to that using SAD or MSE. Besides, its calculation time is reduced to about 50% than the other. In Table 2, the proposed GME method using 3-Parameter Model/GT/SEA is compared with that of [4]. The data illustrate that our method is more accurate and faster than method in [4]. Table 3 gives the runtime of each step of the proposed method. Since some procedures are so fast that can hardly be measured singly, they are combined to record the total time. (As shown in column 2, the processing of Gradient calculation and Gradient thresholding are both very fast, and in column 4, ILSE calculation is also very fast.) From Table 3, we can see that most time is consumed during motion field calculation using SEA (average 68.7%). So the speed of SEA decides the speed of GME, and our contribution on reducing the computational cost of SEA come into effect.

6 Conclusion

In this paper, a fast method is proposed to estimate global motions based on a 3-parameter linear model. While the new model uses less parameter to describe and estimate global motion, its estimation results are still accurate adequately. Using the proposed motion model, a modified ILSE method is presented to estimate the parameters more reasonably by reducing the influence of local motions and false estimations. Besides, another two approaches are proposed to accelerate the procedure of motion field calculation: First, an approach for block judgment is presented. Before motion field calculation, gradient information is analyzed to judge the blocks, by which the number of blocks for subsequent calculation is reduced. Second, a fast algorithm is proposed to compute optical flow, based on both gradient information and intensity information. Extensive experiments show the effectiveness of this technique.

Acknowledgements

This research is sponsored by France Telecom R&D, and the National Natural Science Foundation of China (Grant No. 60135020, 60475010 and 60121302).

References

1. Tekalp, A., ed.: Digital Video Processing. Prentice Hall, Beijing (1998)
2. Rath, G., Makur, A.: Iterative least squares and compression based estimations for a four-parameter linear global motion model and global motion compensation. IEEE Trans. on Circuits & Systems for Video Technology **9** (1999) 1075–1099
3. Jain, J., Jain, A.: Displacement measurement and its application in interframe image coding. IEEE Trans. on Communications **29** (1981) 1799–1808
4. Sorwar, G., Murshed, M., Dooley, L.: Fast global motion estimation using iterative least-square technique. 4th International Conference on Information, Communications & Signal Processing and 4th IEEE Pacific-Rim Conference On Multimedia (2003)
5. Davis, L.: A survey of edge detection techniques. Computer Graphics and Image Processing **4** (1975) 248–270
6. Huang, Y., Chien, S., Hsieh, B., Chen, L.: Global elimination algorithm and architecture design for fast block matching motion estimation. IEEE Trans. on Circuits & Systems for Video Technology **14** (2004) 898–907
7. Li, W., Salari, E.: Successive elimination algorithm for motion estimation. IEEE Trans. on Image Processing **4** (1995) 105–107
8. Gao, X., Duanmu, C., Zou, C.: A multilevel successive elimination algorithm for block matching motion estimation. IEEE Trans. on Image Processing **9** (2000) 501–504
9. Brunig, M., Niehsen, W.: Fast full-search block matching. IEEE Trans. on Circuits & Systems for Video Technology **11** (2001) 241–247
10. Chen, Y., Huang, Y., Fuh, C.: Fast block matching algorithm based on the winner-update strategy. IEEE Trans. on Image Processing **10** (2001) 1212–1222

Kernel-Based Robust Tracking for Objects Undergoing Occlusion*

R. Venkatesh Babu[1], Patrick Pérez[2], and Patrick Bouthemy[2]

[1] Indian Institute of Science, Bangalore, India
venkatesh.babu@gmail.com
[2] IRISA/INRIA, Rennes, France
{perez, Patrick.Bouthemy}@irisa.fr

Abstract. Visual tracking has been a challenging problem in computer vision over the decades. The applications of Visual Tracking are far-reaching, ranging from surveillance and monitoring to smart rooms. Occlusion is one of the major challenges that needs to be handled in tracking. In this work, we propose a new method to track objects undergoing occlusion using both sum-of-squared differences (SSD) and color-based mean-shift (MS) trackers which complement each other by overcoming their respective disadvantages. The rapid model change in SSD tracker is overcome by the MS tracker module, while the inability of MS tracker to handle large displacements is circumvented by the SSD module. Mean-shift tracker, which gained more attention recently, is known for tracking objects in a cluttered environment. Since the MS tracker relies on the global object parameters such as color, the performance of the tracker degrades when the object undergoes partial occlusion. To avoid the adverse effect of this global model, we use the MS tracker so as to track the local object properties instead of a global one. Further a likelihood ratio weighting is used for SSD tracker to avoid drift during partial occlusion and to update the MS tracking modules. The proposed tracker outperforms the traditional MS tracker, as illustrated in the instances applied.

1 Introduction

Visual tracking in a cluttered environment remains one of the challenging problems in computer vision for the past few decades. Various applications like surveillance and monitoring, video indexing and retrieval require the ability to faithfully track objects in a complex scene involving appearance and scale change. Though there exist many techniques for tracking objects, color-based tracking with kernel density estimation, introduced in [1, 2], has recently gained more attention among research community due to its low computational complexity and its robustness to appearance change. The former is due to the use of a deterministic gradient ascent (the "mean shift" iteration) starting at location

* This work was supported by an ERCIM post-doctoral fellowship at IRISA/INRIA, Rennes, France.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 353–362, 2006.

in previous frame. The latter relies on the use of a global appearance model, usually in terms of colors, as opposed to very precise appearance models such as pixel-wise intensity templates.

Though mean-shift tracker performs well on sequences with relatively small object displacement, its performance is not guaranteed for objects in highly cluttered environment especially when it undergoes partial occlusion. In this paper, we try to improve the performance of mean-shift tracker when the object undergoes partial/full occlusion and large displacements. The problem with large displacement is tackled by cascading an SSD tracker with the mean-shift tracker. In order to improve the performance of MS tracker, in the event of the object undergoing partial occlusion, many elementary MS modules(tracking points) are embedded within the object, rather than relying on a single global MS tracker representing the whole object. We also try to improve the performance of MS tracker against large scale changes due to camera operation.

For each of these problems, solutions have been considered so far within pure MS trackers: incorporation of a dynamic model (e.g., using Kalman filter in [1, 3] or particle filter in [4, 5]) to cope with large displacements, occlusions and, to some extent, with scale changes; simple linear histogram updates with fixed forgetting factor [5] for on-line adaptation of reference model; rather complex procedures [6, 7] for addressing the generic problem of scale changes (immaterial of their origin).

The novelty of the proposed approach is to address the problems within a one-step simple approach which exploits the fact that the reference color model and instantaneous motion estimation based on pixel-wise intensity conservation, complement one another. The latter is provided by greedy minimization of the intensity sum-of-squared differences (SSD), which is classic in point tracking and motion field estimation by block matching. Scale changes of the object that are due to the camera zoom effect or ego-motion, are estimated by approximating the dominant apparent image motion by an affine model.

2 Proposed Algorithm

In this work tracking is done in Kalman filter framework. The object to be tracked is specified by location of its center and scale (for a fixed aspect ratio) in the image plane. The objective of the tracking algorithm is to find the correct location of the object in the future frames. An SSD tracker based on frame-to-frame appearance matching, is useful in finding the location of the objects in the future frames. However, the problem with the SSD tracker is its short-term memory which can cause drifting problems or even complete losses in worse cases. On the other hand, MS trackers which rely on persistent global object properties such as color, can be much more robust to detailed appearance changes due to shape and pose changes. This MS tracker has problems with large displacements and its tracking ability is questionable when the object undergoes partial occlusion. It would be efficient if we could combine the advantages of the aforementioned two trackers. In this work, we cascade the two trackers to get a better tracking performance. The measurement obtained by this combined tracker module is used for estimating the states of the Kalman filter.

The state-space representation of the tracker used in Kalman filter framework is given below:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ x_t \\ y_t \\ s_{t+1} \end{bmatrix} = \begin{bmatrix} 2 & 0 & -1 & 0 & 0 \\ 0 & 2 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ x_{t-1} \\ y_{t-1} \\ s_t \end{bmatrix} + \mathbf{w}_t \tag{1}$$

where $\mathbf{x}_t = (x_t, y_t)$ indicates the location of the object center at time t, s_t represents the scale at time t and $\mathbf{w}_t$ is white Gaussian noise with diagonal variance Q. The measurement equation relates the states and measurements at time t as follows:

$$\begin{bmatrix} u_t \\ v_t \\ \xi_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ x_{t-1} \\ y_{t-1} \\ s_t \end{bmatrix} + \mathbf{z}_t \tag{2}$$

where $\mathbf{u}_t = (u_t, v_t)$ is the measured velocity (displacement) of the object, ξ_t is the measured scale at time t, and $\mathbf{z}_t$ is a white Gaussian noise with diagonal variance R. The displacement measurement $\mathbf{u}_t$ is obtained through the SSD-MS tracker module, whereas scale measurement is provided by global parametric motion estimation. The overview of the proposed system is illustrated in Fig. 1. The following subsections explain each of the modules in detail.

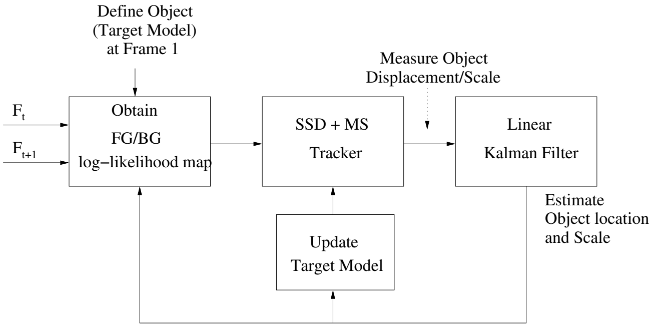


Fig. 1. Overview of the proposed tracking system

2.1 Object-Background Separation and Initialization of Tracking Points

Tracking an object undergoing partial occlusion will be efficient if we could separate precisely the object region from the background at each time instant. This

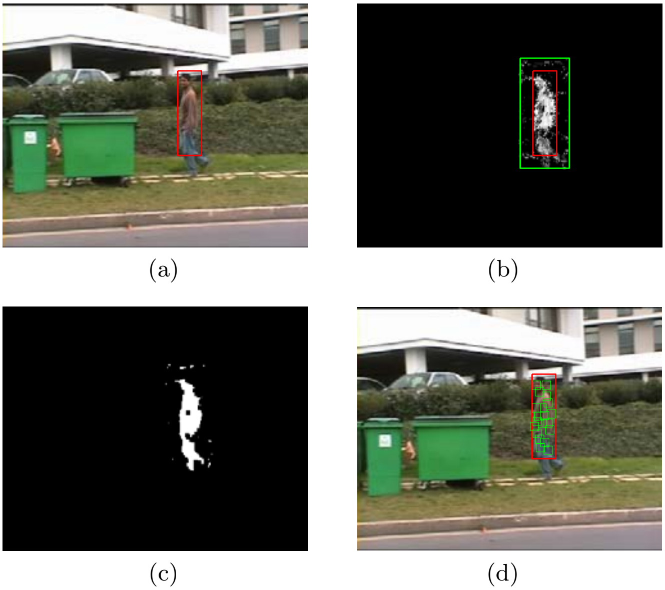


Fig. 2. Track point initialization using T_0: (a) Initial frame with object boundary (b) likelihood map T_0 (c) Mask obtained after morphological operations (d) tracking points with the support region. Here, number of tracking points is 20 with support region of 10^2 pixels.

object-background separation is useful in weighting the pixels for SSD tracker and it helps to locate the reliable MS modules for updating. To achieve this, the R-G-B based joint pdf of the object region and of a band of region surrounding the object region is obtained. This process is illustrated in Fig. 2. The region within the red rectangle is used to obtain the object pdf and the region between the green and red rectangle is used for obtaining the background pdf. Then the resulting log-likelihood ratio of foreground/background region is used to determine object pixels. The log-likelihood of a pixel considered, at time t, within the outer bounding rectangle (green rectangle in Fig. 2) is obtained as

$$L_t(i) = \log \frac{\max\{h_o(i), \epsilon\}}{\max\{h_b(i), \epsilon\}} \tag{3}$$

where $h_o(i)$ and $h_b(i)$ are the probability of ith pixel belonging to the object and background; and ϵ is a small value to avoid division by zero. The non-linear log-likelihood maps the multimodal object/background distribution as positive values for colors associated with foreground and negative values for background. Only reliable object pixels are used as weighting factor for SSD tracker. The weighting factor T_t is obtained as:

$$T_t(i) = \begin{cases} L_t(i) & \text{if} \quad L_t(i) > th_o \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where, th_o is the threshold to decide on the most reliable object pixels. Once the object is localized, with the help of user interaction or by detection in the first frame, the tracking points are placed on the object in the first frame. Then the likelihood map of the object/background is obtained using (9). A binary mask corresponding to T_t is obtained by mapping all positive values of T_t to 1 . This object mask is further subjected to morphological closing operation and used for embedding the tracking points (see Fig. 2). The tracking points are randomly spread, taking care to see that their center lies on the pixels of the object mask. In our experiments the support region of all the tracking point is a square region having side length of $c \cdot \min(\textit{object length}, \textit{object width})$. The typical range of c used in our experiments is 0.3 to 0.5.

2.2 SSD-MS Motion Measurement

The SSD tracker localizes the object in the given search window of the next frame based on minimum distance between the target and candidate object images. SSD tracker works well even for large displacements as long as the object appearance changes only slightly between the two consecutive frames. In reality, the appearance of the object often changes considerably with time. In a typical SSD tracker, the winning candidate becomes the new target for the next time instant. This process makes the SSD forget the original model rapidly with time though for a given target it performs well between any two consecutive frames.

Given the state estimate $(\hat{\mathbf{x}}_{t-1}, \hat{s}_{t-1})$ at previous instant, the SSD-based displacement estimate

$$\begin{aligned}\mathbf{u}_t^{ssd} = \arg\min_{\mathbf{u}\in W} \sum_{\mathbf{d}\in D} T_t . [F_t(\mathbf{u} + \hat{\mathbf{x}}_{t-1} + \hat{s}_{t|t-1}\mathbf{d}) \\ - F_{t-1}(\hat{\mathbf{x}}_{t-1} + \hat{s}_{t-1}\mathbf{d})]^2\end{aligned} \tag{5}$$

where T_t is the weighting function obtained from the foregroung/background likelihood maps. F_{t-1} and F_t are the two consecutive intensity images, $\hat{s}_{t|t-1} = \hat{s}_{t-1}$ is the scale prediction, W is the search window, and D is the normalized subimage support (rectangle same as object-size, with its origin being the object-center).

In our work instead of using a single MS tracker for the entire object, we use multiple small regions of the object for tracking. The locations of these tracking points are randomly placed on the object area with the help of the previously obtained object/background likelihood map.

This first displacement estimate given by (5) is used for initializing these mean-shift trackers. Let N be the total number of tracking points. The target color models $\mathbf{q}^n = (q_i^n)_{i=1\cdots m}$, with $\sum_{i=1}^m q_i^n = 1$, are composed of m bins in some appropriate color space (e.g., RGB or Hue-Saturation, in our experiments RGB color space with 10 bins along each dimension is used), where superscript $n \in N$ indicates the nth model corresponding to the nth tracking point. It is gathered at the initialization of the overall tracking. The candidate histogram $\mathbf{p}^n$, at location $\mathbf{x}^n$ and scale s in the current frame is given by:

$$p_i^n(\mathbf{x}^n) = \frac{\sum_{\mathbf{d}\in s\cdot D} k(s^{-2}|\mathbf{d}|^2)\delta[b(\mathbf{x}^n + \mathbf{d}) - i]}{\sum_{\mathbf{d}\in s\cdot D} k(s^{-2}|\mathbf{d}|^2)} \tag{6}$$

where $k(x)$ is a convex and monotonic decreasing kernel profile, almost everywhere differentiable and with support D, which assigns smaller weights to pixels far away from the center (in our experiments Epanechnikov kernel profile is used), δ is the Kronecker delta function, and function $b(\mathbf{x}) \in \{1...m\}$ is the color bin number at pixel $\mathbf{x}$ in the current frame. One seeks the location whose associated candidate histogram is as similar as possible to that of the target one. When similarity is measured by Bhattacharya coefficient $\rho(\mathbf{p}^n, \mathbf{q}^n) = \sum_i \sqrt{p_i^n q_i^n}$, convergence towards the nearest local maximum is obtained by the iterative mean-shift procedure [8]. In our case, this gradient ascent at time t is initialized at $\mathbf{y}_0^n = \hat{\mathbf{x}}_{t-1}^n + \mathbf{u}_t^{ssd}$ and proceeds as follows:

1. Given current location $\mathbf{y}_0^n$ compute histogram $\mathbf{p}^n(\mathbf{y}_0^n)$ and Bhattacharya coefficient $\rho(\mathbf{p}^n(\mathbf{y}_0^n), \mathbf{q}^n)$.
2. Compute candidate position

$$\mathbf{y}_1^n = \frac{\sum_{\mathbf{d}\in s\cdot D} w^n(\mathbf{y}_0^n + \mathbf{d})k'(s^{-2}|\mathbf{d}|^2)(\mathbf{y}_0^n + \mathbf{d})}{\sum_{\mathbf{d}\in s\cdot D} w^n(\mathbf{y}_0^n + \mathbf{d})k'(s^{-2}|\mathbf{d}|^2)}$$

   with weights at location $\mathbf{x}$

$$w^n(\mathbf{x}) = \sum_{i=1}^{m} \sqrt{\frac{q_i^n}{p_i^n(\mathbf{y}_0^n, s)}}\delta[b(\mathbf{x}) - i].$$

3. while $\rho(\mathbf{p}^n(\mathbf{y}_1^n, s), \mathbf{q}^n) < \rho(\mathbf{p}^n(\mathbf{y}_0^n, s), \mathbf{q}^n)$
   do $\mathbf{y}_1^n \leftarrow \frac{1}{2}(\mathbf{y}_1^n + \mathbf{y}_0^n)$
4. if $\|\mathbf{y}_1^n - \mathbf{y}_0^n\| < \varepsilon$ stop
   otherwise set $\mathbf{y}_0^n \leftarrow \mathbf{y}_1^n$ and repeat Step 2.
5. Use only the reliable displacements out of N measurements for the final estimate. Let $\mathcal{R} \subset \{\mathbf{y}^1 \dots \mathbf{y}^N\}$ be the set of all reliable MS trackers. The final motion estimate is obtained as: $\mathbf{y} = mean(\mathbf{y}^i), \quad i \in \mathcal{R}$.

The final estimate provides the displacement estimate $\mathbf{u}_t = \mathbf{y} - \hat{\mathbf{x}}_{t-1}$. In our experiment the MS trackers whose Bhattacharya coefficients lie in the top 10 percent are considered as reliable MS trackers. Finally, the two entries associated to this measurement in the covariance matrix R_t of the observation model (2) are chosen as

$$\sigma_u^2 = \sigma_v^2 = e^{\alpha(1-\mathrm{mean}\{\rho_i\})}, \quad i \in \mathcal{R} \tag{7}$$

where ρ_i are the Bhattacharya coefficients of the reliable MS trackers. The parameter α set to 25 in the experiments.

2.3 Scaling Measurement

Scaling is another important parameter in visual tracking. Often the scale change of the objects are due to the camera zoom operation or camera ego-motion. The scale change in our work is measured (to be plugged in Kalman Filter) through the affine motion parameters of the global (dominant) image motion between the current and next frame. Such parameters can be estimated in a fast and robust way [9]. If 2×2 matrix A_t stands for the linear part of the affine motion model thus estimated at time t, the scale measurement is

$$\xi_t = \xi_{t-1} \left\{ 1 + \frac{\mathrm{trace}(A_t)}{2} \right\}. \tag{8}$$

2.4 Algorithm Summary

The complete algorithm is summarized below. Given previous reference color models $\mathbf{q}_{t-1}^n$ and previous state estimate $(\hat{\mathbf{x}}_{t-1}, \hat{s}_{t-1})$ with error covariance P_{t-1}:

1. Obtain the likelihood map T_t of the object/background according to (9)
2. Obtain SSD-based displacement measurement $\mathbf{u}_t^{ssd}$ according to (5) with the weighting factor (T_t).
3. Correct this measurement with reliable MS trackers, initialized at $\mathbf{u}_t^{ssd}$ and with reference color models $\mathbf{q}_{t-1}^n$, to obtain final measurement $\mathbf{u}_t$.
4. Estimate global affine motion over the image and derive new scale measurement ξ_t according to (8).
5. Using displacement and scale measurement $\mathbf{u}_t$ and ξ_t, update state estimate with Kalman filter, providing $(\hat{\mathbf{x}}_t, \hat{s}_t)$ and associated error covariance P_t.

Initial state $(\hat{\mathbf{x}}_1, \hat{s}_1 = 1)$ in frame 1 is obtained either by manual interaction or by detection, depending on the scenario of interest.

3 Results and Discussion

The proposed algorithm has been tested on several videos and it has been observed to have performed well, not only under partial, but also brief full occlusion. The tracking result for 'walk' sequence is shown in Fig. 4 for both proposed tracker and the SSD+ global MS tracker. In this sequence the object undergoes

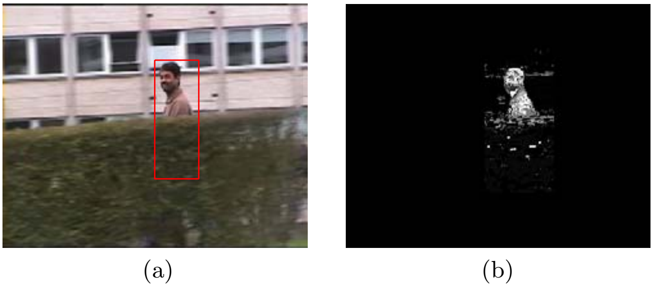

(a) (b)

Fig. 3. (a) One frame showing object under partial occlusion and the (b) corresponding log-likelihood map

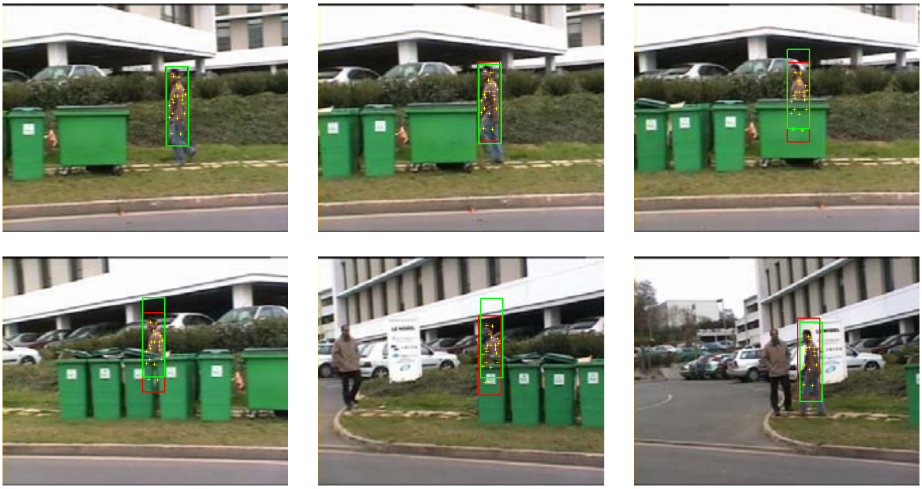


Fig. 4. Tracking result of proposed system against the SSD+global MS on 'walk' sequence for frames 2,20,40,100,140 and 200 are shown. The '+' marks indicate the MS tracking points, the red rectangle corresponds to the proposed tracking and the green rectangle corresponds to the SSD+global MS tracker result.

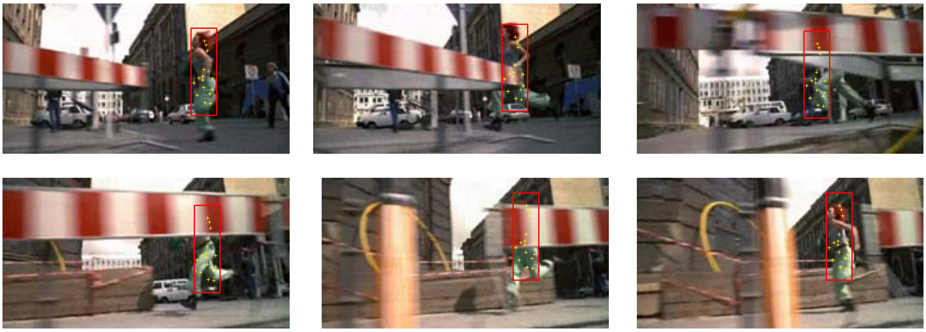


Fig. 5. Tracking result of proposed system on a movie sequence 'run-lola-run' is shown. The '+' marks indicate the MS tracking points, the red rectangle corresponds to the proposed tracking result.

a partial occlusion. The proposed system was able to track the object correctly without any shift when the object is partly occluded. The global MS based tracker undergoes a large shift during partial occlusion. The tracking result of the proposed algorithm for a dynamic video shot from the movie 'run-lola-run' is shown in Fig. 5. The number of MS tracking points used in 'walk' sequences were 20 and 15 MS tracking points were used for 'lola' sequence. The presented walk video sequences were shot with a hand-held cam-coder, which automatically adjusts the brightness based on the background environment. The change of object color in these videos are due to this automatic adjustment of camera

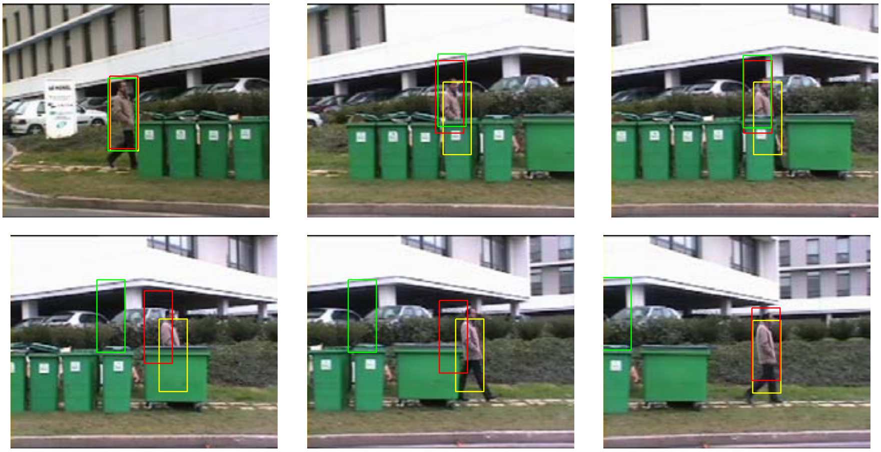


Fig. 6. Tracking result of proposed system against the SSD+global MS on another 'walk' sequence is shown. The yellow rectangle corresponds to to the proposed tracking with model update, the red rectangle corresponds to the proposed tracking without model update and the green rectangle corresponds to the SSD+global MS tracker result.

parameters. In such videos, keeping a fixed color model will drastically reduce the accuracy of tracking. The person in 'walk' sequence shown in Fig. 6 not only goes through partial occlusion, but the object luminance undergoes drastic change from the starting frame to the final frame. In such case, it is necessary to adapt the tracking model to brightness/color change. In our system, the tracking points whose support lie mostly on the object region are updated with the latest color model. The area of the intersection between the object and the support of each tracking point is estimated using the recently obtained log-likelihood map. Fig. 3 shows the log-likelihood map when the object undergoes partial occlusion. The model corresponding to a particular tracking point is updated if the object area occupies a certain minimal area (in our system its set as 50%) of its support region. Only the support regions of the tracking points that intersect with the object region are used for updating the target color model of the corresponding tracking points. Let M_t be the binary mask obtained from the log-likelihood map:

$$M_t(i) = \begin{cases} 1 & \text{if} \quad L_t(i) > th_u \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

In our experiments, th_u is set as 2 for considering only the most reliable object pixels. Let $\mathbf{q}^n$ be one of the MS models located at $\mathbf{x}$ with support region $s \cdot D + \mathbf{x}$. If $\frac{1}{|D|} \sum_{d \in D} M_t(s \cdot D + \mathbf{x}) > D_{th}$, then replace the model $\mathbf{q}^n$ with the recent model obtained as:

$$q_i^n(\mathbf{x}) = \frac{\sum_{\mathbf{d} \in D} M_t(s \cdot D + \mathbf{x}) k(s^{-2}|\mathbf{d}|^2) \delta[b(\mathbf{x} + \mathbf{d}) - i]}{\sum_{\mathbf{d} \in D} M_t(s \cdot D + \mathbf{x}) k(s^{-2}|\mathbf{d}|^2)} \tag{10}$$

In our experiments, D_{th} is set as 0.5 (corresponding to 50% of the support area).

The results obtained with such update model is shown in Fig. 6. In this example the SSD + global MS tracker fails to track the object till the end of the sequence. The proposed method without model update tracks the object till the end of sequence but there has been some drift from the object due to the luminance/color change of the object. The proposed tracker with model update is able to track the object with out any drift.

4 Conclusion

In this paper, we have proposed an efficient visual tracker by coupling SSD and mean-shift algorithm, which have complementary properties. By tracking local color properties of the object using multiple MS tracking points on the object, instead of a single global MS tracker, improves the performance when the object undergoes partial occlusion. The better performance of the proposed tracker over combined SSD, global mean-shift tracker is shown using various video sequences. Since both trackers have real-time computational complexity, the proposed compound tracker is suitable for real time tracking of objects.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. Conf. Comp. Vision and Pattern Recog., Hilton Head, SC (2000)
2. Bradski, G.: Computer vision face tracking as a component of a perceptual user interface. In: Workshop on App. of Comp. Vision, Princeton, NJ (1998)
3. Zhu, Z., Ji, Q., Fujimura, K.: Combining kalman filtering and mean shift for real time eye tracking under active ir illumination. In: Proc. Int. Conf. Pattern Recognition, Quebec City, Canada (2002)
4. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Proc. Europ. Conf. Computer Vision, Copenhagen, Denmark (2002)
5. Nummiaro, K., Koller-Meier, E., Van Gool, L.: An adaptive color-based particle filter. Image and Vision Computing **21** (2003) 99–110
6. Collins, R.: Mean-shift blob tracking through scale space. In: Proc. Conf. Comp. Vision Pattern Rec., Madison, Wisconsin (2003)
7. Zivkovic, Z., Kröse, B.: An EM-like algorithm for color-histogram-based object tracking. In: Proc. Conf. Comp. Vision Pattern Rec., Washington, DC (2004)
8. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Machine Intell. **25** (2003) 564–577
9. Odobez, J.M., Bouthemy, P.: Robust multiresolution estimation of parametric motion models. J. Visual Com. Image Repr. **6** (1995) 348–365

Adaptive Object Tracking with Online Statistical Model Update

KaiYeuh Chang and Shang-Hong Lai

Dept. of Computer Science, National Tsing Hua University,
Hsinchu 300, Taiwan
{kaiyeuh, lai}@cs.nthu.edu.tw

Abstract. In this paper, we propose a statistical model-based contour tracking algorithm based on the Condensation framework. The models include a novel object shape prediction model and two statistical object models. The object models consist of the grayscale histogram and contour shape PCA models computed from the previous tracking results. With the incremental singular value decomposition (SVD) technique, these three models are learned and updated very efficiently during tracking. We show that the proposed shape prediction model outperforms the affine predictor through experiments. Experimental results show that the proposed contour tracking algorithm is very stable in tracking human heads on real videos with object scaling, rotation, partial occlusion, and illumination changes.

1 Introduction

Visual tracking has been a main focus of research in video analysis and processing. With the rapid growth of digital video in consumer electronics and video surveillance, reliable visual tracking techniques have been strongly demanded recently. Previous methods on visual tracking can be divided into the model based [1-3] and non-model based [4-6] approaches. For objects with well-defined models, the corresponding object tracking problem is easier. However, the requirement of constructing object models beforehand limits the practical feasibility of this approach, especially the object may undergo a wide variety of different motions, including 3D rigid and non-rigid motions. Non-model based tracking approaches treat object tracking as an optimization problem. They normally track objects from image sequences by using the latest tracking result as reference for the object. This approach is sensitive to error drift, i.e. error accumulation. Once the object is lost during tracking, it may not be found again.

There are many different modifications for the particle filter. Rui and Chen [8] modified the way for computing the posterior probability by considering the current image in the prediction phase. Recently, Maggio and Cavallaro [9] combined the particle filter and mean shift techniques for refined object tracking. Okuma et al. [10] proposed an algorithm that integrates the particle filter and adaboost techniques for tracking multiple targets. Like mean shift, Nummiaro et al. [11] employed the Bhattacharyya coefficient in the color distribution for object tracking in a particle filter framework.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 363 – 372, 2006.

Besides, Jepson et al. [2] models the appearance of the object under tracking via three models. The wandering model reliably estimates the parameters for rapid temporal variations and shorter temporal histories, the stable model captures the behavior of temporally stable image observations, and the last model accounts for data outliers. With online EM algorithm, they update the models adaptively to achieve robust object tracking.

In this paper, we propose a visual tracking algorithm based on the framework of the Condensation algorithm [1] for tracking object contour via on-line object model generation and dynamic prediction model update. The Condensation, or particle filter, framework consists of the prediction and measurement phases. The sample contours at the current frame are predicted during the prediction phase. In the measurement phase, the probability of each sample is computed from the image information at the current frame. The pre-trained model and prediction matrix (or motion model) used in the Condensation technique are learned from the best results achieved by using the Kalman filter tracking. This two-pass method (one for learning and the other for tracking) is inconvenient for tracking general objects in practice.

Recently, Lim et al. [3] proposed a method on self constructing and updating model for appearance-based object tracking. However, contour tracking provides a more detailed object tracking result, not only the position, rotation, and scale but also the object shape deformation. In most appearance-based tracking algorithms, the object is represented by a rectangle or an ellipse which can be aligned by a simple transformation. The entire information inside the rectangle or the ellipse can be exploited to determine the tracking result. Nonetheless, it is difficult to use the entire image region information for deformable contour tracking since it requires establish point-to-point correspondences between two deformable regions, especially for the particle filter which uses many random samples to approximate the probability distribution.

In this paper, we propose an object contour tracking algorithm based on the particle filter framework. It only needs an initial contour at the first frame and then the object models and the prediction matrix are constructed online from the previous contour tracking results automatically. In the proposed algorithm, we build two online models for the target object – one is the shape model and the other is the grayscale histogram model. The grayscale histogram simply records the grayscale information inside the object contour region. Each of these two models is represented by a mean vector and several principle components, which are adaptively computed with the incremental singular value decomposition technique [3,7].

1.1 Condensation Framework

Here we briefly describe the condensation framework for visual tracking. In the condensation tracking, the prediction phase can be represented as a probability term

$$p(\mathbf{x}_t \mid X_{t-1}) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \tag{1}$$

where $\mathbf{x}_t$ is the predicted state, $\mathbf{x}_{t\text{-}1}$ is the state at the previous time t-1 and $X_{t\text{-}1}$ denotes the whole previous states. The left function means that we use all the previous states to predict the current state. For simplicity, we only take the state at the previous time instant to predict the current state, which is represented as conditional probability on the right hand side of the above equation. On the other hand, the measurement phase can be represented by the following function

$$p(\mathbf{z}_t \mid \mathbf{x}_t), \tag{2}$$

where $\mathbf{z}_t$ is the current observed information. This means that we take the predicted state to see if it matches the current observation. The object tracking problem can be thought of using all the image and object model information that we currently have to find the object, which is given by the following conditional probability

$$p(\mathbf{x}_t \mid Z_t), \tag{3}$$

where Z_t denotes all the information that we currently have. To compute the probability function (3), we employ the Bayes rule as follows

$$p(\mathbf{x}_t \mid Z_t) = k_t p(\mathbf{z}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid Z_{t-1}), \tag{4}$$

where

$$p(\mathbf{x}_t \mid Z_{t-1}) = \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid Z_{t-1})$$

and k_t is a normalization constant. From the above two equations, we will see how the prediction and measurement phases work in the Condensation tracking algorithm.

Now we define several symbols for explaining the Condensation algorithm. The points of each sample contour is arranged into a vector $\mathbf{s}=[x_1\, y_1 \ldots x_i\, y_i \ldots x_n\, y_n]^{\mathbf{T}}$, where n is the total number of the points along the contour and (x_i,y_i) is the coordinate of the i-th point. The symbol $\mathbf{s}_{i,t}$ means the i-th sample contour at the t-th frame. The symbol $\pi_{i,t}$ means the probability of $\mathbf{s}_{i,t}$. Similarly, $\mathbf{s}_{i,t}^{sampled}$ denotes the i-th random sample at the t-th frame according to the probabilities of all sample contours at the $(t$-1)-th frame. We denote the predicted contour of $\mathbf{s}_{i,t}^{sampled}$ as $\mathbf{s}_{i,t}^{pred}$. The Condensation tracking algorithm [1] is given as following:

Given N contour samples and their corresponding probabilities $\{\mathbf{s}_{i,t\text{-}1}, \pi_{i,t\text{-}1}, i=1,\ldots, N\}$ at the $(t$-1)-th frame, we want to find N contour samples and their corresponding probabilities $\{\mathbf{s}_{i,t}, \pi_{i,t}, i=1,\ldots, N\}$ at the t-th frame based on the following procedure:

1. We sample N contours $\mathbf{s}_{i,t}^{sampled}, i=1,\ldots,N$ from the previous sample contours $\mathbf{s}_{i,t\text{-}1}, i=1,\ldots, N$ according to their associated probabilities $\pi_{i,t\text{-}1}$.
2. The prediction function $p(\mathbf{x}_t \mid \mathbf{x}_{t-1} = \mathbf{s}_{i,t}^{sampled})$ is applied to predict $\mathbf{s}_{i,t}^{pred}$ from $\mathbf{s}_{i,t}^{sampled}$.
3. We find the contour $\mathbf{s}_{i,t}$ around the predicted contour $\mathbf{s}_{i,t}^{pred}$ at the t-th frame.
4. Compute $\pi_{i,t} = p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_{i,t})$.
5. The expected contour $\mathbf{s}_t$ at the t-th frame is computed as follows

$$\mathbf{s}_t = \sum\nolimits_i \pi_{i,t} \mathbf{s}_{i,t}\ . \tag{5}$$

At step 3 and 4, we may find some points like edge boundary as a contour found from the frame around the predicted one. For simplicity, we call it image contour. With a shape model, a fitting algorithm can be applied to fit the image contour to see if it is suitable to be the target object's boundary.

In Condensation [1], the image contour is a reference result while the fitted contour is $\mathbf{s}_{i,t}$. When $\mathbf{s}_{i,t}$ is close to the image contour, it is highly possible to be the object's boundary. Taking the fitted contour as $\mathbf{s}_{i,t}$ can reduce the noise from the image and make $\mathbf{s}_{i,t}$ to be a reasonable contour at least. However, when the model is unreliable or lack of information, the fitted contour may not be a good choice. Besides, the model needs the object information. Thus, we take the image contour as $\mathbf{s}_{i,t}$ and the fitted contour as the reference result. This decision is with lots of risk and choosing the contour points from the image becomes very important.

2 Visual Tracking Based on Condensation Algorithm

The Condensation algorithm requires three basic components in the tracking algorithm. The first is the prediction function (step 2), the second is the method to find $\mathbf{s}_{i,t}$ around $\mathbf{s}_{i,t}^{pred}$ according to the information of the current frame (step 3), and the last is the way to compute the probability $\pi_{i,t} = p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_{i,t})$ (step 4).

Assume that we already have the prediction matrix, the object models and a way to find the image contour. We show how the prediction function works and how to measure the probability here. The determination of the prediction matrix, the object models and the image contour will be described in the later sections.

Our prediction function is given as following

$$\mathbf{s}_{i,t}^{pred} = P\begin{bmatrix} \mathbf{s}_{i,t}^{sampled} \\ 1 \end{bmatrix}, \tag{6}$$

where P is the prediction matrix and the 1 below $\mathbf{s}_{i,t}^{sampled}$ is used to model the global translation of the contour. In our experiment, the performance of the prediction matrix is degraded when the contour has a larger change in the shape or the position than the previous ones. Thus, we combine $\mathbf{s}_{i,t}^{sampled}$ and $\mathbf{s}_{i,t}^{pred}$ to get a more stable prediction result.

$$\mathbf{s}_{i,t}^{pred} \leftarrow C_{pred}\mathbf{s}_{i,t}^{pred} + \left(1 - C_{pred}\right)\mathbf{s}_{i,t}^{sampled}, \tag{7}$$

where C_{pred} is the weight of the prediction result. Then we add Gaussian noises into the rotation, scale and translation parameters, i.e.

$$\begin{bmatrix} \mathbf{s}_{i,t}^{pred}\left(2\times j\right) \\ \mathbf{s}_{i,t}^{pred}\left(2\times j+1\right) \end{bmatrix} \leftarrow RS\begin{bmatrix} \mathbf{s}_{i,t}^{pred}\left(2\times j\right) \\ \mathbf{s}_{i,t}^{pred}\left(2\times j+1\right) \end{bmatrix} + \mathbf{t}, \tag{8}$$

where $(\mathbf{s}_{i,t}^{pred}(2\times j), \mathbf{s}_{i,t}^{pred}(2\times j+1))$ is the coordinate of the j-th point of the contour, R and S are the random rotation and scaling matrices and $\mathbf{t}$ is the random translation vector.

To compute the observation probability, we build the grayscale histogram and shape models for the target object and update the object models online. The grayscale histogram is normalized with the total number of pixels inside the contour region, thus representing the occurrence frequency of each bin. For each contour $\mathbf{s}_{i,t}$, we compute the grayscale histogram $\mathbf{h}_{i,t}$ inside it and align it to the mean shape and the aligned result is $\mathbf{s}_{i,t}^{aligned}$.

We project the contour $\mathbf{s}_{i,t}^{aligned}$ and the corresponding grayscale histogram $\mathbf{h}_{i,t}$ onto the object model, which consists of the grayscale histogram and shape PCA models, to compute the reconstructed contour $\mathbf{s}_{i,t}^{recon}$ and grayscale histogram $\mathbf{h}_{i,t}^{recon}$, respectively. Note that the PCA models for the grayscale histogram and the object shape can be adaptively updated from the previous object tracking results by using the incremental singular value decomposition [3, 7]. This PCA reconstruction is used to measure how well the observation fits to the object model. The discrepancies of the model fitting are given as follow:

$$dh_{i,t} = \left\| \mathbf{h}_{i,t} - \mathbf{h}_{i,t}^{recon} \right\| = \sum\nolimits_j \left| \mathbf{h}_{i,t}\left(j\right) - \mathbf{h}_{i,t}^{recon}\left(j\right) \right|, \tag{9}$$

and

$$ds_{i,t} = \left\| \mathbf{s}_{i,t}^{aligned} - \mathbf{s}_{i,t}^{recon} \right\| = \frac{\sum\nolimits_j \left(\mathbf{s}_{i,t}^{aligned}\left(j\right) - \mathbf{s}_{i,t}^{recon}\left(j\right)\right)^2}{\left| \mathbf{s}_{i,t} \right|}. \tag{10}$$

Thus, the observation probability function is defined by

$$\pi_{i,t} = p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_{i,t}) \propto \exp\left(-C_s ds_{i,t} - C_h dh_{i,t}\right), \tag{11}$$

where C_s and C_h are two constants to represent the weights of the two factors. Thus, we compute the conditional probability $\pi_{i,t} = p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_{i,t})$ as follows:

$$\pi_{i,t} = p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_{i,t}) = \frac{\exp\left(-C_s ds_{i,t} - C_h dh_{i,t}\right)}{\sum\nolimits_j \exp\left(-C_s ds_{j,t} - C_h dh_{j,t}\right)}. \tag{12}$$

3 Shape Prediction Matrix

Consider two object contours $\mathbf{s}_{t\text{-}1}$ and $\mathbf{s}_t$ at two connective frames t-1 and t. We employ a prediction matrix P to describe their relationship as follows:

$$P \begin{bmatrix} \mathbf{s}_{t-1} \\ 1 \end{bmatrix} \sim \mathbf{s}_t \tag{13}$$

When considering N consecutive frames, the problem of estimating the prediction matrix P turns to minimizing the following energy function

$$\sum_{i=2}^{N} \left\| \mathbf{s}_i - P \begin{bmatrix} \mathbf{s}_{i-1} \\ 1 \end{bmatrix} \right\|^2 . \tag{14}$$

Let $\mathbb{S}_{i,j} = \begin{bmatrix} \mathbf{s}_i & \mathbf{s}_{i+1} & \cdots & \mathbf{s}_j \\ 1 & 1 & \cdots & 1 \end{bmatrix}$ and $S_{i,j} = \left[\mathbf{s}_i \ \ \mathbf{s}_{i+1} \ \ \cdots \ \ \mathbf{s}_j \right]$. Then we can compute the matrix P by using the least square estimation, thus leading to

$$P = S_{2,N} \mathbb{S}_{1,N-1}^{\mathrm{T}} \left(\mathbb{S}_{1,N-1} \mathbb{S}_{1,N-1}^{\mathrm{T}} \right)^{-1}. \tag{15}$$

Note that the prediction matrix P needs to be updated dynamically during object tracking to better describe the shape deformation. However, we do not need to store all the previous contours to compute the prediction matrix. Instead, we can simply store and update the two matrixes $S_{2,N}\mathbb{S}_{1,N-1}^{\mathrm{T}}$ and $\mathbb{S}_{1,N-1}\mathbb{S}_{1,N-1}^{\mathrm{T}}$ for updating the prediction matrix. When new contours at M frames are available for prediction matrix update, these two matrixes become

$$S_{2,N+M}\mathbb{S}_{1,N+M-1}^{\mathrm{T}} = S_{2,N}\mathbb{S}_{1,N-1}^{\mathrm{T}} + S_{N+1,N+M}\mathbb{S}_{N,N+M-1}^{\mathrm{T}} \tag{16}$$

and

$$\mathbb{S}_{1,N+M-1}\mathbb{S}_{1,N+M-1}^{\mathrm{T}} = \mathbb{S}_{1,N-1}\mathbb{S}_{1,N-1}^{\mathrm{T}} + \mathbb{S}_{N,N+M-1}\mathbb{S}_{N,N+M-1}^{\mathrm{T}}\,. \tag{17}$$

If each contour is composed of n points, we need to store the two matrices of sizes $2n$×$(2n+1)$ and $(2n+1)$×$(2n+1)$, respectively. However, computing the inverse of the matrix $\mathbb{S}_{1,N-1}\mathbb{S}_{1,N-1}^{\mathrm{T}}$ is computationally expensive for large n. The problem is even worse when frequent prediction matrix update is required in practice. Here we propose an efficient way to update the prediction matrix P by using the incremental SVD technique as described below.

Considering N frames for estimating the prediction matrix P, we arrange all N contours column by column as follows:

$$P\mathbb{S}_{1,N-1} = S_{2,N} \tag{18}$$

Using singular value decomposition (SVD) to decompose the matrix $\mathbb{S}_{1,N-1}$ yields

$$\mathbb{S}_{1,N-1} \underset{SVD}{=} U_{1,N-1}\Sigma_{1,N-1}V_{1,N-1}^{\mathrm{T}} \tag{19}$$

Then the prediction matrix P can be computed by

$$P = S_{2,N}V_{1,N-1}\Sigma_{1,N-1}^{-1}U_{1,N-1}^{\mathrm{T}}\,. \tag{20}$$

The size of the matrix $V_{1,N\text{-}1}$ will increase with the data or frame number. By combining $S_{2,N}$ and $V_{1,N\text{-}1}$ into a matrix $S_{2,N}V_{1,N\text{-}1}$, we only need to maintain the three matrices $U_{1,N\text{-}1}$, $\Sigma_{1,N\text{-}1}$ and $S_{2,N}V_{1,N\text{-}1}$. If n points compose a contour and the k largest eigenvalues with the corresponding eigenvectors of the SVD are needed, then the three matrix sizes are $(2n+1)$×k, k×k and $2n$×k, respectively.

The incremental SVD technique can be used to easily generate and update the matrices $U_{1,N\text{-}1}$ and $\Sigma_{1,N\text{-}1}$. For the computation of the matrix $S_{2,N}V_{1,N\text{-}1}$, its computational complexity only depends on the total number of the kept eigenvalues in the SVD. The incremental SVD produces a matrix V after a step of SVD. This matrix V is used to update $V_{1,N\text{-}1}$. Assume that new M data arrives and we keep $k_{1,N\text{-}1}$ and $k_{1,N+M\text{-}1}$ eigenvalues before and after the model update. Note that the size of V is $(k_{1,N\text{-}1}+M)$×$k_{1,N+M\text{-}1}$. We can divide V into two parts, i.e. $V = \begin{bmatrix} V_{up}^{\mathrm{T}} & V_{bottom}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, where the sizes of V_{up} and V_{bottom} are $k_{1,N\text{-}1}$×$k_{1,N+M\text{-}1}$ and M×$k_{1,N+M\text{-}1}$, respectively. The matrix $V_{1,N+M\text{-}1}$ becomes $V_{1,N+M-1} = \begin{bmatrix} (V_{1,N-1}V_{up})^{\mathrm{T}} & V_{bottom}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. Thus, we update $S_{2,N}V_{1,N\text{-}1}$ as follow

$$S_{2,N+M}V_{1,N+M-1} = \begin{bmatrix} S_{2,N}V_{1,N-1} \end{bmatrix} V_{up} + S_{N+1,N+M}V_{bottom}\,. \tag{21}$$

4 Contour Refinement

The image contour is composed of several nodal points. The main idea to find the nodal points is to search a large gradient in the normal direction of each point in $\mathbf{s}_{i,t}^{pred}$ since there is usually a large gradient in the object boundary. In addition, there are several criteria to be considered for the nodal points. Firstly, the directions of the gradient and the normal line should be as consistent or adverse as possible. Secondly, we compute an average distance according to the large gradient criterion. Then, the distance between the nodal point and the corresponding one in the predicted contour $\mathbf{s}_{i,t}^{pred}$ is assumed to be close to the average distance. Thirdly, if more than one point meets the above two criteria, we set all of them to be candidates for the nodal point. Thus, the nodal points are selected based on the score function given as follows:

$$Score\left(\mathbf{p}^{feature}\right) = \begin{cases} random\left(minScale, maxScale\right)\times \\ \left|\mathbf{n}\bullet\mathbf{g}^{feature}\right|\exp\left(-\dfrac{\left|\left\|\mathbf{p}^{feature}-\mathbf{p}^{pred}\right\|-avgDis\right|}{C_{var}}\right), \text{if } \dfrac{\left|\mathbf{n}\bullet\mathbf{g}^{feature}\right|}{\left\|\mathbf{g}^{feature}\right\|} > C_{angle} \\ 0, \text{otherwise} \end{cases}, \quad (22)$$

where $\mathbf{p}^{pred}$ is one of the points of $\mathbf{s}_{i,t}^{pred}$, $\mathbf{n}$ is a unit vector for the corresponding normal direction, $\mathbf{p}^{feature}$ is one of the points located on the normal line of $\mathbf{p}^{pred}$, and $\mathbf{g}^{feature}$ is the image gradient at the location $\mathbf{p}^{feature}$. Thus, $\left|\mathbf{n}\bullet\mathbf{g}^{feature}\right|$ is the amount of the gradient projected on the normal direction. The condition $\left|\mathbf{n}\bullet\mathbf{g}^{feature}\right|/\left\|\mathbf{g}^{feature}\right\| > C_{angle}$ means the first criterion and C_{angle} is the threshold of the minimum cosine value of the angle between $\mathbf{n}$ and $\mathbf{g}^{feature}$. The function *random(minScale, maxScale)* returns a random number between *minScale* and *maxScale*. Both *minScale* and *maxScale* are positive values. This random number generation is used to implement the third criterion. The function $\exp\left(-\left|\left\|\mathbf{p}^{feature}-\mathbf{p}^{pred}\right\|-avgDis\right|/C_{var}\right)$ is used to implement the second criterion, where the parameter C_{var} controls the distance closeness.

The n-th point of $\mathbf{s}_{i,t}^{pred}$ is denoted by $\mathbf{p}_{n}^{pred}$, $\mathbf{p}_{m,n}^{feature}$ is the m-th candidate point for $\mathbf{p}_{n}^{pred}$, and $\mathbf{p}_{n}^{feature}$ is the n-th nodal point of the contour $\mathbf{s}_{i,t}$. Thus, $\mathbf{p}_{n}^{feature}$ is determined based on maximizing the score function as follows

$$\mathbf{p}_{n}^{feature} = \arg\max_{\mathbf{p}_{m,n}^{feature}}\left(Score\left(\mathbf{p}_{m,n}^{feature}\right)\right). \quad (23)$$

5 Experimental Results

In this section, we show some experimental results on video tracking by using the modified Condensation tracking algorithm with online model adaptation. We also give the experimental results by using the affine motion prediction for comparison with the proposed algorithm that uses the novel shape prediction matrix update scheme. The affine motion model can be represented by the following equation

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} \mathbf{s}_{x,t}(j) \\ \mathbf{s}_{y,t}(j) \end{bmatrix}+\begin{bmatrix} e \\ f \end{bmatrix}=\begin{bmatrix} \mathbf{s}_{x,t+1}(j) \\ \mathbf{s}_{y,t+1}(j) \end{bmatrix}, \tag{24}$$

where ($\mathbf{s}_{x,t}(j)$, $\mathbf{s}_{y,t}(j)$) is the coordinate of the j-th point of the contour at the t-th frame. By assuming the affine motion to be constant in a short period, we can estimate the affine motion parameters by using the least square solution as follows

$$\sum_{\substack{t\in \text{previous several frames} \\ j\in \text{points of a contour}}} \begin{bmatrix} A & C & 0 & 0 & D & 0 \\ C & B & 0 & 0 & E & 0 \\ 0 & 0 & A & C & 0 & D \\ 0 & 0 & C & B & 0 & E \\ D & E & 0 & 0 & F & 0 \\ 0 & 0 & D & E & 0 & F \end{bmatrix}\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \sum_{\substack{t\in \text{previous several frames} \\ j\in \text{points of a contour}}} \begin{bmatrix} G \\ H \\ I \\ J \\ K \\ L \end{bmatrix}, \tag{25}$$

where $A = \mathbf{s}_{x,t}(j)\mathbf{s}_{x,t}(j), B = \mathbf{s}_{y,t}(j)\mathbf{s}_{y,t}(j), C = \mathbf{s}_{x,t}(j)\mathbf{s}_{y,t}(j), D = \mathbf{s}_{x,t}(j),$
$E = \mathbf{s}_{y,t}(j), F = 1, G = \mathbf{s}_{x,t}(j)\mathbf{s}_{x,t+1}(j), H = \mathbf{s}_{y,t}(j)\mathbf{s}_{x,t+1}(j), I = \mathbf{s}_{x,t}(j)\mathbf{s}_{y,t+1}(j),$
$J = \mathbf{s}_{y,t}(j)\mathbf{s}_{y,t+1}(j), K = \mathbf{s}_{x,t+1}(j),$ and $L = \mathbf{s}_{y,t+1}(j).$

In our implementation, the total number of samples in the particle filter is 90. The shape prediction matrix used in our tracking algorithm is initialized to be an identity matrix and it is updated for every 3 frames. For ease of computation, we employ the forgetting factor scheme in the incremental SVD technique [3]. The forgetting factor is set to be (n - 3)/n, where n is the total number of frames used for estimating the shape prediction matrix. The total number of frames used in the experiments is set to 40 empirically. For the affine motion estimation, we use a small number of frames for the least square estimation because there are only 6 affine motion parameters. From our experimental result, the affine predictor is less stable when a certain degree of errors are involved in the tracking result. In contrast, our prediction matrix accounts for more temporal shape variations across more frames, thus making the shape prediction more stable. This is evident from Figure 1 and 2.

In addition, we show the performance of our contour tracking algorithm on two sequences. The total number of samples in the particle filter is set to 90 and we use 40 previous frames for estimating the shape prediction matrix. The data amounts for updating the shape and the grayscale histogram models are both 100.

The first testing video sequence is the Dudek sequence [2], which is about 38 seconds with 15 fps frame rate. In this sequence, we track the contour of the head, which contains different scales, poses, translations and partial occlusions in the video. Some of the tracking results are depicted in Figure 3. The background of the video is cluttered and contains many different objects. The tracking results show that our algorithm generally can provide quite reliable tracking performance.

The second testing video sequence is about 51 seconds with 15 fps frame rate. This sequence contains a person moving in a room and the contour of his head is our target object. The main difficulty is the large illumination changes from dark to bright conditions, which can test our grayscale histogram model. The result shows that the proposed algorithm can track the head contour pretty well for the entire sequence as some frames depicted in Figure 4.

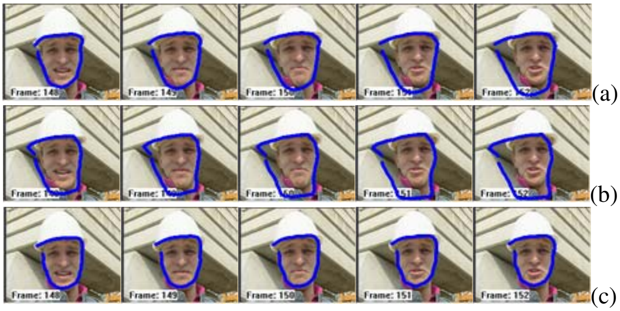


Fig. 1. Predictor comparison I: (a) The tracking results by using the affine predictor for frames 148-152. The previous 2 frames are used to predict the affine matrix. (b) The tracking results by using the affine predictor for frames 148-152. The previous 4 frames are used to predict the affine matrix. (c) The tracking results by using the proposed shape prediction matrix for frames 148-152. The total number of frames used for estimating the shape prediction matrix is 40.

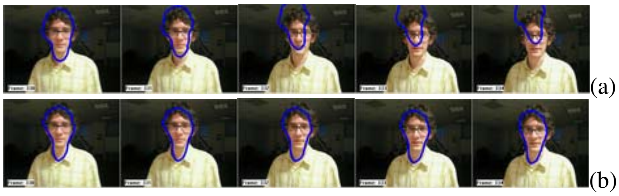


Fig. 2. Predictor comparison II: The tracking results by using (a) the affine predictor (2 previous frames) (b) the proposed shape prediction matrix (40 previous frames) for frames 148-152

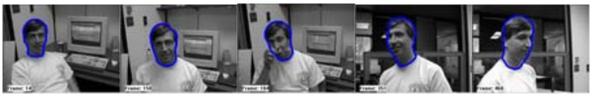


Fig. 3. The tracking results of the Dudek face sequence [2] with different scales, poses, translations and partial occlusions

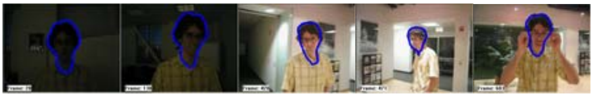


Fig. 4. The tracking results by using the proposed algorithm on the second testing video sequence with different scales, poses, translations, and the significant illumination changes from dark to bright conditions

6 Conclusion

In this paper, we purpose an adaptive contour tracking algorithm based on the Condensation algorithm with online updating the shape prediction matrix and object models. The novel shape prediction model is very flexible and accounts for temporal shape deformation. The object model consists of the grayscale histogram and contour PCA models adaptively computed from previous tracking results by using the incremental SVD technique. The proposed Condensation tracking algorithm with online model update is computationally efficient due to the use of incremental SVD for updating both the object and shape prediction models. Due the online model update capability and the flexible shape prediction model, the proposed tracking algorithm is very stable since most recent tracking results are taken for the model update. Experimental results show the proposed tracking algorithm can track human heads very reliably in cluttered environment under large lighting variations on several real videos.

References

1. Isard, M. and Blake, A.: CONDENSATION - conditional density propagation for visual tracking. International Journal of Computer Vision, Vol. 29. (1998) 5-28
2. Jepson, A. D., Fleet, D. J. and El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Conf. Compute Vision Pattern Recognition, Vol. 1. (2001) 415-422
3. Lim, J., Ross, D., Lin, R.S. and Yang, M.H.: Incremental learning for visual tracking. Neural Information Processing Systems 17 (NIPS 2004)
4. Liu, T.-L. and Chen, H.-T.: Real-time tracking using trust-region methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26. (2004) 397-402
5. Comaniciu, D., Ramesh, V. and Meer, P.: Real-time tracking of non-rigid objects using mean shift. In Proc. Conf. Computer Vision and Pattern Recognition, Vol. 2. (2000) 142-149
6. Comaniciu, D., Ramesh, V. and Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25. (2003) 564-575
7. Levy, A. and Lindenbaum, M.: Sequential Karhunen Loeve basis extraction and its application to image. IEEE Transactions on Image Processing, Vol. 9. (2000) 1371-1374
8. Rui, Y. and Chen, Y.: Better Proposal Distributions: Object Tracking Using Unscented Particle Filter. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2001) 786-793
9. Maggio, E. and Cavallaro, A.: Hybrid Particle Filter and Mean Shift tracker with adaptive transition model. Proc. of IEEE Signal Processing Society International Conference on Acoustics, Speech, and Signal Processing (ICASSP). (2005) 19-23
10. Okuma, K., Taleghani, A., de Freitas, N., Little, J. J. and Lowe, D. G.: A Boosted Particle Filter: Multi-target Detection and Tracking. European Conference on Computer Vision. (2004) 28-39
11. Nummiaro, K., Koller-Meier, E. and Van Gool, L.: An Adaptive Color-Based Particle Filter. International Journal of Image and Vision Computing, Vol. 21. (2003) 99-110

Inducing Semantic Segmentation from an Example

Yaar Schnitman[1], Yaron Caspi[1], Daniel Cohen-Or[1], and Dani Lischinski[2]

[1] Tel Aviv University, Israel
{caspiy, dcor}@tau.ac.il
[2] The Hebrew University of Jerusalem, Israel
danix@cs.huji.ac.il

Abstract. Segmenting an image into semantically meaningful parts is a fundamental and challenging task in computer vision. Automatic methods are able to segment an image into coherent regions, but such regions generally do not correspond to complete meaningful parts. In this paper, we show that even a single training example can greatly facilitate the induction of a semantically meaningful segmentation on novel images within the same domain: images depicting the same, or similar, objects in a similar setting.

Our approach constructs a non-parametric representation of the example segmentation by selecting patch-based representatives. This allows us to represent complex semantic regions containing a large variety of colors and textures. Given an input image, we first partition it into small homogeneous fragments, and the possible labelings of each fragment are assessed using a robust voting procedure. Graph-cuts optimization is then used to label each fragment in a globally optimal manner.

1 Introduction

Image segmentation, the process of identifying homogeneous regions in an image, is a fundamental task in a large number of applications in image and video processing. A particularly challenging instance of image segmentation is the problem of automatically identifying semantically meaningful regions in an image. This problem is often referred to as *image labeling*, since its goal is to associate each pixel in the image with a label denoting a semantically meaningful part.

While the objective of grouping pixels according to color, texture, and other cues has been dealt with in many ways, the challenge of aggregating pixels into segments representing meaningful parts is much harder. This is due to the fact that such parts are often too complex to be characterized using low-level image features, such as color or texture. Furthermore, the semantic interpretation of an image is highly subjective, depending on both the application, and the user. For example, while some applications are concerned with separating a person from the background, others might require the partitioning of a person's body into its various parts, as demonstrated in Figure 1.

In this paper, we present a novel labeling method, which computes a semantically meaningful partitioning of an input image, as induced from one (or more) correctly segmented training image. Both the input and the training image(s)

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 373–384, 2006.

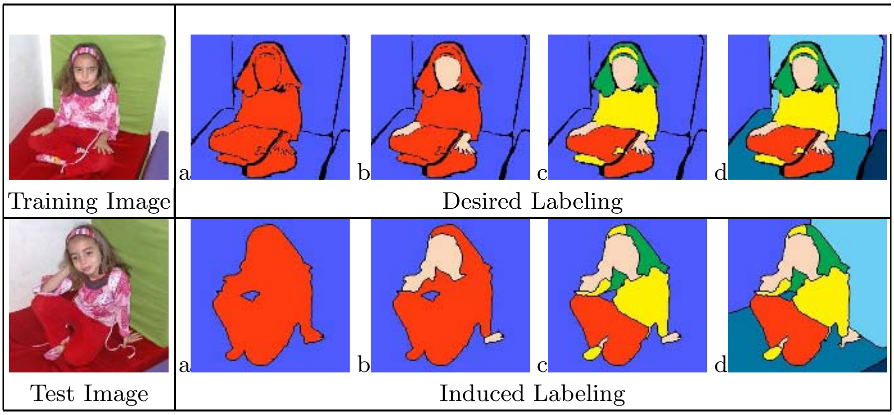


Fig. 1. Inducing different semantically meaningful segmentations. This figure illustrates how four different labelings are induced. A single training labeling is provided each time, all with respect to the train image in left upper corner. Labeling (a) is a binary partitioning between foreground and background. Labeling (b) also distinguishes between skin and clothes. Labeling (c) decomposes the figure into hair and clothes, while labeling (d) breaks up the background into several parts. Note that in general, the various parts cannot be characterized by common image space attributes, and they cannot be inferred without an explicit description or an example.

are assumed to be from the same domain: having similar illumination, resolution and scale characteristics, and depicting similar scenes. The meaningful parts in the training image are recognized in the input image, and the correct assignment of pixels into labels is induced. Such a mechanism is required in various applications, like removal, replacement, or recoloring of a certain object in a series of images. For example, one might want to change the color of a garment worn by a model in all the photographs taken during a particular session.

Our method constructs a *non-parametric* model of the provided training pair by selecting a set of patch-based representatives inside each labeled region in the training image. These representatives are used to quantify the degree of resemblance between small regions in the input image and the labeled regions in the training set. This simple, yet informative representation, which is derived directly from the image, has proved its worth in other applications, such as texture synthesis [1], image analogies [2], recoloring [3], and image and video completion [4, 5]. Here we extend this approach to image labeling.

Image analogies and texture transfer [2, 6] are a general framework by which various types of filters are learned from a single unfiltered and filtered image pair, and induced on novel images. As mentioned above, these methods gain their strength from a simple patch-based sampling scheme. The method we present has a similar flavor and shares their simplicity. However, the former methods cannot induce labeling since the decisions they make are inherently local. In contrast, our method makes use of a global optimization step for finding an optimal pixel labeling.

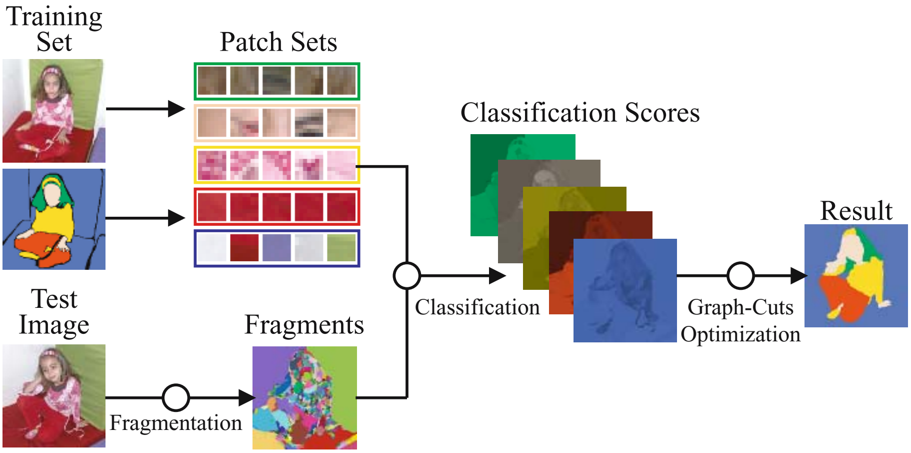


Fig. 2. An overview of our method: The labeled training image is sampled, creating sets of square patches, one set for each label. Given an input image it is first over-segmented into a collection of small homogeneous fragments. Assignment costs are then computed for each fragment-label pair. Finally, a graph-cuts multi-label optimization is used to find the globally optimal labeling of the fragments.

In this work, we assume that small homogeneous regions always belong to the same semantic part of the image. Hence, we over-segment the input image into *fragments*, small arbitrarily-shaped and simply-connected pixel clusters, and compute the labeling at the fragment level. The fragmentation has a profound effect on the final result, as it enforces a locally coherent labeling and facilitates a voting scheme as a means for a robust per fragment label assignment. Furthermore, working at the fragment level reduces the computational complexity and improves performance.

Figure 2 outlines our method. Patch sampling is performed over the labeled training image, defining a set of patches, each representing a labeled region of the image. During labeling induction, the input image is first partitioned into small fragments. Then, assignment cost is computed for every fragment-label pair. Next, these costs together with additional contiguity constraints are incorporated into a graph-cuts multi-label optimization, to yield a global labeling of the fragments. The combination of patch-based sampling, fragmentation, and the graph-cuts optimization results in a segmentation scheme that incorporates both local and global information, allowing effective induction of semantically meaningful labelings from one image to another.

2 Background and Related Work

Segmentation is a well-studied problem. A common approach for segmentation aggregates local cues such as color, texture, edges or various filter responses, by which pixels are clustered into contiguous, homogeneous regions (e.g, [7, 8]. For a survey of segmentation methods, see [9].

While these methods are successful in clustering image pixels into homogeneous regions, they cannot automatically group the resulting clusters into semantically meaningful parts. However, they do provide natural image building blocks, or image fragments, which can facilitate various region based decisions, such as label assignments. We argue that determining whether an entity belongs to a particular semantic part is more easily done at the fragment level, than on a pixel-by-pixel basis.

The limitations of a pixel level decision are also addressed by global methods. By global methods we refer to methods that formulate the problem as a minimization problem over the space of labelings/segmentations. The feasibility of the global approaches is bounded by the exponential complexity of the space of all possible solutions. Therefore, different algorithms restrict the space in order to make the minimization tractable. The restrictions are usually formulated with priors, such as continuity or smoothness. They yield a minimization of an error function comprised of two error terms: the data constraint, and a pairwise constraint. Examples of global methods include normalized cuts [10], belief propagation [11], and graph-cuts [12], which is used in this paper.

Another limitation of previous segmentation methods is the descriptive power of the parametric model that they use to represent a segment, e.g., distribution of colors, textures or some other features. A powerful alternative is to use examples as an implicit representation. Example-based non-parametric modeling avoids the complications of parametric modeling. This approach has been applied successfully in applications ranging from texture synthesis to image completion [1, 2, 6, 13, 3, 4].

An example based representation is also used for detection and segmentation of objects from a specific class [14]. There, the task is to segment an object in an image, based on a large set of pre-segmented images, all from the same family (e.g., horses). In contrast, we are interested in labelings induced by as few as a single example. The image building blocks used in their method are also termed fragments. However, their fragments are rectangular tiles of variable size, while in our work, fragments may have an arbitrary shape determined by the context of the image.

Segmentation is also closely related to the problem of extracting objects from images. Because the task is so challenging, interactive solutions were developed, where the user assists the segmentation process. In particular, graph-cuts optimization has proved to be an effective tool for interactive image segmentation [15, 16]. The optimization is used to find segmentations, which are consistent with color, edges, and the user defined constraints. Graph-cuts have been extended to handle multiple (more than two) segment problems, using the alpha-expansion algorithm [17]. Recent works on video tooning [18] and rotoscoping [19] are related to our work. They also face the problem of producing a consistent segmentation for a sequence of similar images. Their approach takes advantage of frame coherence, computing 3D clusters of pixels in the space-time video volume. The user then outlines the semantic regions using a rotoscoping interface. We are also interested in segmenting similar images, but make no assumptions regarding coherence among the images, and identify semantic regions automatically based on a small training set.

3 Algorithm

In this section we describe our algorithm for inducing the labeling of the training image onto the test image. Let I_{train} denote the training image and L_{train} the labeling of its pixels by k different labels. Given an input (test) image I_{test} our goal is to compute its corresponding labeling L_{test}. We begin by describing how patches in I_{train} are used to compute labeling costs for pixels in I_{test} (Sec. 3.1). Rather than attempting to label each individual pixel in I_{test} we partition it into small homogeneous fragments (Sec. 3.2) and compute more robust labeling costs for each fragment (Sec. 3.3). Finally, we use graph-cuts optimization to assign a label to each fragment in a globally optimal manner (Sec. 3.4).

3.1 Pixel Labeling Costs

Given I_{train} and L_{train} we create a patch-based classifier by representing each label by a set of square patches, sampled from the corresponding region in I_{train}. We get k such sets $\{S_l\}_{l=1}^{k}$, one for each label. Each set contains a variable number of patches, depending on the number of pixels with that label in I_{train}. All patches are of uniform size $m \times m$, which is chosen beforehand so it is proportional to the scale of details in the image, such as $m = 7$ or $m = 20$. Figure 3 depicts the representation of each segment class by a set of sampled patches. Next, we define $\varphi(p, l)$ to be the cost of assigning label l to a pixel $p \in I_{test}$. Informally, a low cost $\varphi(p, l)$ indicates that there is a high likelihood that p should be labeled with l, and vice versa. We compute $\varphi(p, l)$ by matching P, the $m \times m$ square patch centered at p, with the patches in the set S_l. The cost is proportional to the distance to the nearest neighbor of P within S_l:

$$\varphi(p, l) = \min_{P' \in S_l} \frac{ssd(P, P')}{M},$$

where $ssd(P, P')$ is the sum of squared distances between the patches P and P', both treated as M-length vectors, where $M = m \times m \times 3$ in the case of three RGB color channels.

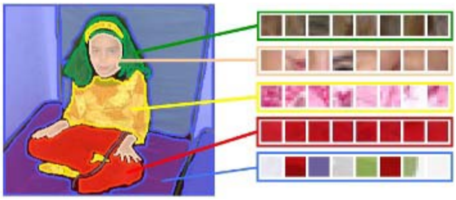


Fig. 3. Patch-based classifier. Each semantic part is represented by a set of square patches, sampled from within the corresponding region in the training image.

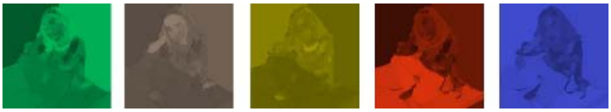


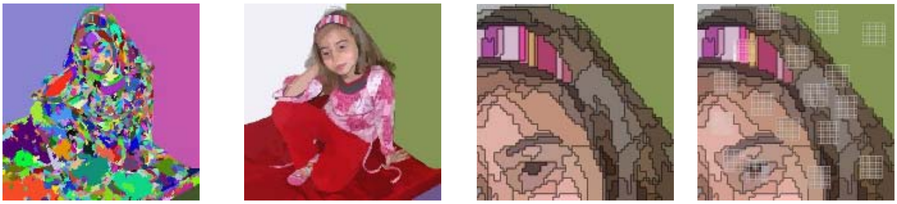
Fig. 4. Visualization of fragment labeling costs. Costs range in the interval [0,1] and are colorized according to each label's representative color, as defined in the Figure 3.

3.2 Fragmentation

The search for the nearest-neighboring patches within each set S_l is computationally intensive. In order to reduce the number of such searches, we partition I_{test} into small, color-homogeneous regions, which we refer to as *fragments*. These fragments are arbitrarily-shaped and may contain from a few pixels to thousands of pixels. We exploit the resulting structure to accelerate the algorithm by evaluating the labeling costs only for a small fraction of the pixels within each fragment, and then use voting to arrive at a set of labeling costs for each fragment.

The fragmentation is performed such that fragments are smaller in more detailed areas of I_{test}, and larger in more homogeneous regions. In addition, it is important that fragment boundaries align with edges in the image, since such edges may correspond to the boundary between different semantic regions. Fragments which comply to these criteria may be computed using mean-shift segmentation [8] with sufficiently small kernel bandwidths. Figure 5 demonstrates the result of fragmentation. Notice how small fragments form in highly detailed areas (such as the hair and shirt regions), while large fragments form in homogeneous areas (such as the walls in the background).

In addition to reducing the computational cost, fragmentation actually helps produce better results, for two reasons. First, fragmentation constrains pixels within the same fragment to be assigned to the same label, thereby enforcing a locally coherent labeling. Second, the voting procedure performed on pixels within each fragment produces more robust labeling costs.



random colorization mean value colorization detailed close-up representative patches

Fig. 5. Fragmentation. The input image is fragmented into arbitrarily-shaped homogeneous regions, which we call fragments. Fragment sizes vary according to the amount of detail in various image areas, and their boundaries are aligned with edges in the image. The label assignment of each fragment is computed by choosing representative patches.

3.3 Fragment Labeling Costs

We apply a voting scheme in order to compute the labeling costs of each fragment. For each fragment $f \in I_{test}$ we pick a few representative pixels:

$$Rep(f) = \{p_i \in f\}_{i=1}^{R_f},$$

where R_f is proportional to the number of pixels in f, for example: $R_f = \lfloor\sqrt{|f|}\rfloor$. Figure 5 visualizes fragments along with their representative pixels (and the corresponding patches). The cost of assigning label l to fragment $f \in I_{test}$ is defined as:

$$\varphi(f, l) = \text{median}\left\{\varphi(p, l) | p \in Rep(f)\right\}.$$

Choosing the median value is a robust voting scheme, which is insensitive to outliers. By the end of this process, each fragment is associated with k different costs, one for each label. Figure 4 shows a visualization of the labeling costs that were computed for the example in Figure 1.

As patches and fragment dimensions are frequently similar, it is often the case that the patches centered at the representative pixels contain pixels outside the fragment, affecting the fragment's labeling costs. A simple solution would be to introduce weights into the computation of the distance between patches, but this interferes with the efficient nearest neighbor search that our implementation currently employs. It should be noted however that the effect of these outliers is significantly reduced by the voting scheme.

3.4 Graph-Cuts Optimization

After all pixels in the test image I_{test} have their labeling costs, we need to find L_{test}, the globally optimal labeling. A label assignment that minimizes the total labeling cost and also is devoid of small, disconnected segments. Thus, we also require the labeling to be consistent with the presence (or absence) of edges in I_{test}.

In order to satisfy these requirements, we add an additional pairwise constraint $\psi\left(p, q, L(p), L(q)\right)$ between each pair of neighboring pixels $\langle p, q\rangle$. This constraint enforces label assignments to change only across evident edges in I_{test}. The constraint $\psi(p, q, L(p), L(q))$ is 0 when the labels assigned to p and q are the same ($L(p) = L(q)$) and otherwise, proportional to the evidence of $\langle p, q\rangle$ not being an edge in I_{test}. Specifically,

$$\psi\left(p, q, L(p), L(q)\right) = \begin{cases} 0 & L(p) = L(q) \\ 1 - \nabla(p, q) & \text{otherwise} \end{cases} \tag{1}$$

where $\nabla(p, q)$ is the difference (in RGB distance) between pixels p and q, attenuated and scaled to the range $[0, 1]$. Furthermore, we enforce the restriction that pixels within each fragment should be labeled the same, in order to reduce the combinatorial search-space and achieve a satisfactory approximation at reduced computational costs. This is implemented the by specifing our energy term $E(L)$ in terms of fragments instead of pixels:

$$E(L) = \sum_f |f| \cdot \varphi\left(f, L(f)\right) + \alpha \sum_{\langle f_1, f_2\rangle} \psi\left(f_1, f_2, L(f_1), L(f_2)\right).$$

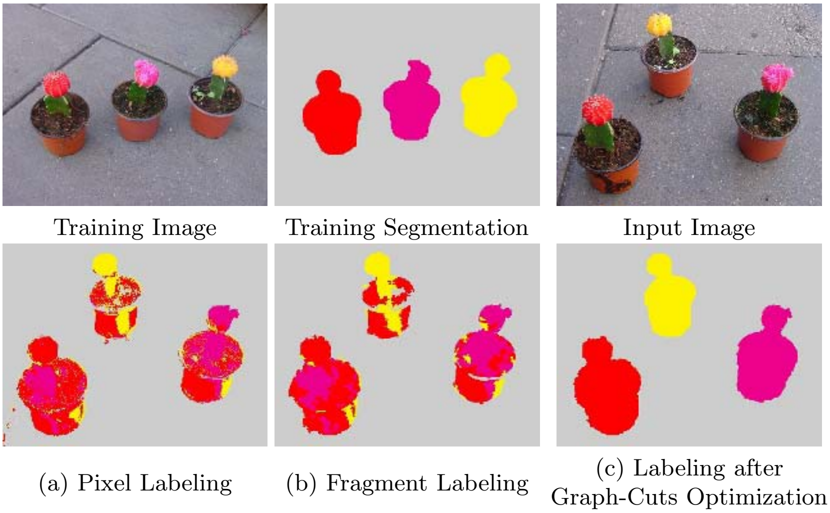


Fig. 6. The contribution of fragmentation and global optimization. The training set consists of four semantically meaningful segments: three plants and the background. Notice that the plants' segments have very similar local characteristics, except in their upper part, which has a unique color. (a) shows that a direct labeling of pixels fails to induce a locally coherent segmentation, due to the close similarity. (b) shows that labeling of fragments produces coherent labeling, but the labeling is over-segmented. (c) shows that a global combinatorial optimization captures semantically meaningful parts, and assigns the correct label.

Here $\langle f_1, f_2\rangle$ are neighboring fragments in I_{test}. $\varphi\left(f, L(f)\right)$ is the cost defined in Sec. 3.3, weighted by the size of each fragment. The pairwise constraint $\psi()$ is extended to neighboring fragments by summing the constraint over their shared boundary:

$$\psi\left(f_1, f_2, L(f_1), L(f_2)\right) = \sum_{\langle p,q\rangle, p\in f_1, q\in f_2} \psi\left(p, q, L(f_1), L(f_2)\right).$$

Finally, L_{test} is determined by solving: $L_{test} = \min_L E(L)$. We apply the graph-cuts multi-label optimization technique for the fragment-based energy term $E(L)$, using the alpha-expansion method [12].

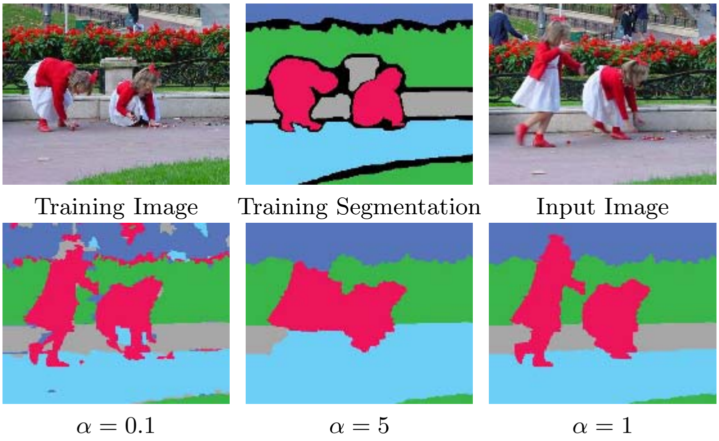
4 Implementation and Results

Image fragmentation is implemented with the mean-shift algorithm from [20]. Graph-cuts optimization is implemented with the *Maxflow* algorithm from [21], which computes the optimal cut for each alpha-expansion move. In this implementation the trade-off between regions and boundaries, is controlled by a single parameter α. Figure 7 demonstrates the profound effect of this parameter on the results. In all our experiments we used a fixed α value for all the images within the same series, typically setting α to one or a nearby value. For searching square patches we uses a kd-tree [22]. In most of our results, we use patches of size 7×7. To reduce computation time, we sample only 5% of possible patches within each label in the training pair. Labeling of images of size 256×256 pixels, with three to six labels takes a few seconds on a 1.8 GHz Pentium 4 machine.

We test our method in the following scenario: Within a set of similar images, one image is chosen to be the training image. We manually segmented the image into multiple semantically meaningful parts, and colored each part with a unique color. Ambiguous pixels were marked in black. Trained by this image pair, our algorithm is used to induce the correct labeling on the remaining images. By image similarity we require that all images should depict the same subject (e.g., birds on the grass), have similar illumination conditions and are of similar resolution and scale. In some of the examples, we apply manual histogram equalizations and scaling in order to enforce these requirements.

Depending on the application, there are many ways to segment a particular image into semantically meaningful parts. Figure 1 depicts our experiments of creating different conceivable labelings and their induction on another image within the same domain. Note that certain semantically meaningful labelings, like the one that merges clothes and hair under the same label, cannot be characterized in terms of simple image features, and thus cannot be inferred without an explicit description or an example.

As described above, we use fragmentation to enforce locally-coherent labeling of pixels, and graph-cuts optimization to induce the globally optimal assignment of labels to fragments. Particularly, propagation of information across fragments is crucial in scenarios where different semantic parts share similar sub-parts. We


Training Image

Training Segmentation

Input Image

$\alpha = 0.1$

$\alpha = 5$

$\alpha = 1$

Fig. 7. The tradeoff between fragment labeling costs and the pairwise smoothness constraints is controlled by a single parameter α. A low α value favors boundaries and produces a over-segmentation, while a high α value penalizes boundaries, producing under-segmentation.

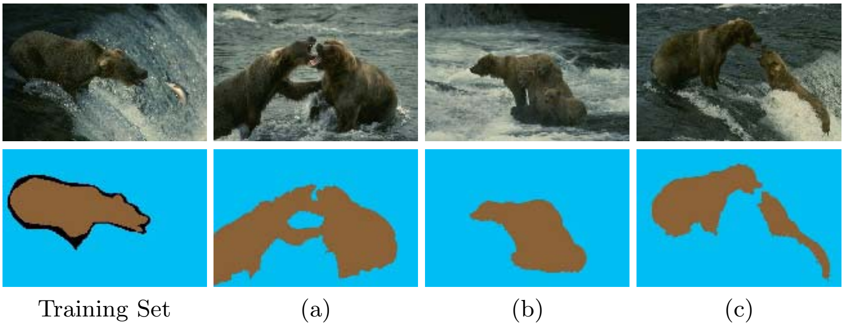


Fig. 8. Arbitrary segment shapes. The segmentation between bear and water is induced on three different images. Notice that the induced segmentation may contain holes (a), and be non-contiguous (b), but our method cannot separate multiple objects belonging to the same label (c).

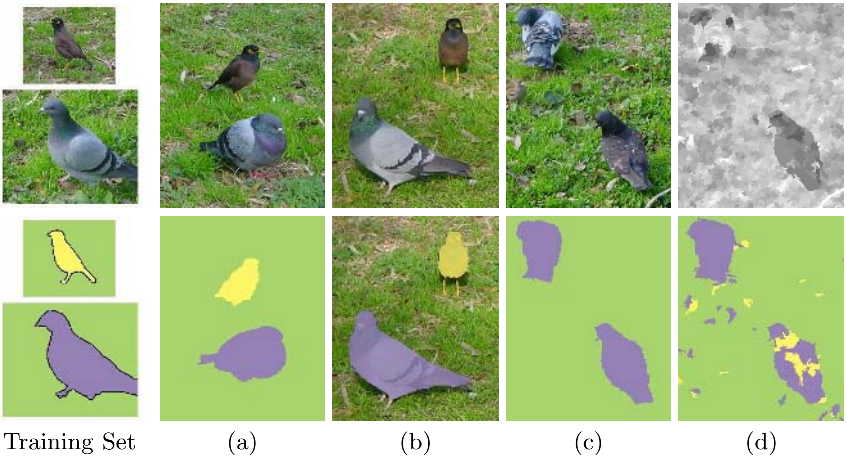


Fig. 9. Object detection and identification. Images of two types of birds are given as a training set, each bird marked by a distinct label. Labeling results (a-c) demonstrate our algorithm's ability to detect the presence of each bird. The gray scale image (d) demonstrates the labeling assignment costs of image (c), disclosing a greater confidence over the labeling of the left bird than the right bird, as the latter differ from the training image. Results without graph cut optimization (d) illustrate its contribution.

demonstrate the effect of the fragmentation and global optimization in Figure 6, by showing the consequences of omitting each of them.

Our method is invariant to the number of instances of each semantic part within the image, and insensitive to the shape of each part. Figure 8 shows the labeling of parts with different topology, in particular, holes (b) and discontinuities (c). On the other hand, our method cannot separate segments which correspond to multiple instances of the same semantic label, as in the bear family image (c).

The ability to segment an image and detect the semantic meaning of each part is demonstrated in Figure 9. Images of two types of birds are given as a training set, where each bird is marked by a distinct label. The results demonstrate the ability to correctly detect and distinguish between the birds (a). The bottom image in (b) also demonstrates that since fragments respect image edges, the labeled regions have correct boundaries, which agree with the underlying image.

Note that the lower right bird in (c) top constitutes a difficult case, since it is a bit darker than its counterpart in the example image, making it more similar to the second type of bird. This is evident in the gray-scale figure (d) top, which visualizes the optimal cost of the globally optimal labeling, demonstrating the problem of making clear cut decision. This image can be treated as a confidence map, and it discloses a greater confidence over the labeling of the left bird than the right bird.

5 Discussion and Future Work

"The whole is greater than the sum of its parts" [23] is one of the Gestalt principles. In this paper, we identify the parts (fragments) of the whole (meaningful object) by assigning them a common label. In general, labeling meaningful parts is known to be a difficult task. We have shown that inducing a labeling from an example can effectively perform this task for a set of images from the same domain. We can attribute this to the following reasons: (i) The example defines the granularity of the desired output. That is, whether we expect to label a complete human body, or its sub-parts: hands, torso, head, etc. (ii) The example allows the use of a non-parametric model to alleviate the huge space of parts. These have more discriminative properties than parametric models. Figure 6 demonstrates that applying the labeling to fragments rather than pixels provides better results. Note that the shapes of our fragments are data dependent rather than being predefined (e.g., rectangles or ellipses). We believe that the labeling problem should address meaningful building blocks, and that pixels are too small to be informative.

In the future we would like to investigate the applicability of our method to a series of images with some spatial coherence. Such coherence can assist the labeling of fragments across the images by considering their relative spatial position in the image. This can then lead to various tracking methods applicable to video with scenarios which include occlusions and frequent scene cuts.

References

1. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: International Conference on Computer Vision, Corfu, Greece (1999) 1033–1038
2. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. Computer Graphics and Interactive Techniques (2001) 327–340
3. Welsh, T., Ashikmin, M., Mueller, K.: Transferring color to greyscale images. Computer Graphics and Interactive Techniques (2002) 277–280

4. Drori, I., Cohen-Or, D., Yehurun, H.: Fragment-based image completion. ACM Transactions on Graphics, (SIGGRAPH) (2003) 303–312
5. Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: IEEE Conference on Computer Vision and Pattern Recognition. (2004) 120–127
6. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. ACM Transactions on Graphics, (SIGGRAPH) (2001) 341–346
7. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. Trans. on Pattern Analysis and Machine Intelligence **13** (1991) 583–598
8. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. Trans. on Pattern Analysis and Machine Intelligence (2002) 603–619
9. Lucchese, L., Mitra, S.K.: Color image segmentation: A state-of-the-art survey. Proc. Indian National Science Academy (INSA-A) **67** (2001) 207–221
10. Shi, J., Malik, J.: Normalized cuts and image segmentation. Trans. on Pattern Analysis and Machine Intelligence **22** (2000) 888–905
11. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. Exploring artificial intelligence in the new millennium (2003) 239–269
12. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: International Conference on Computer Vision, Vancouver , BC. (2001) 105–112
13. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. IEEE Comput. Graph. Appl. **22** (2002) 56–65
14. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: European Conference on Computer Vision. Volume 2., Copenhagen, Denmark (2002) 109–124
15. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. ACM Transactions on Graphics, (SIGGRAPH) **23** (2004) 303–308
16. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics **23** (2004) 309–314
17. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23** (2001) 1222–1239
18. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. ACM Transactions on Graphics, (SIGGRAPH) **23** (2004) 574–583
19. Agarwala, A., Hertzmann, A., Salesin, D., Seitz, S.: Keyframe-based tracking for rotoscoping and animation. ACM Transactions on Graphics, (SIGGRAPH) **23** (2004) 584–591
20. Christoudias, C.M., Georgescu, B.: Edge detection and image segmentation (edison) system. (http://www.caip.rutgers.edu/riul/research/robust.html)
21. Boykov, Y., Kolmogorov, V.: Maxflow software. (http://www.cs.cornell.edu/People/vnk/software.html)
22. Mount, D., Arya, S.: Ann: Library for approximate nearest neighbor searching. (http://www.cs.umd.edu/ mount/ANN/)
23. Wertheimer, M.: Productive Thinking. Collins, NY (1945)

Super Resolution Using Graph-Cut

Uma Mudenagudi, Ram Singla, Prem Kalra, and Subhashis Banerjee

Department of Computer Science and Engineering,
Indian Institute of Technology Delhi,
Hauz Khas, New Delhi 110016, India
{uma, pkalra, suban}@cse.iitd.ernet.in
ram.singla@gmail.com

Abstract. This paper addresses the problem of super resolution - obtaining a single high-resolution image given a set of low resolution images which are related by small displacements. We employ a reconstruction based approach using MRF-MAP formalism, and use approximate optimization using graph cuts to carry out the reconstruction. We also use the same formalism to investigate high resolution expansions from single images by deconvolution assuming that the point spread function is known. We present a method for the estimation of the point spread function for a given camera. Our results demonstrate that it is possible to obtain super-resolution preserving high frequency details well beyond the predicted limits of magnification.

1 Introduction

In this paper we investigate the problem of obtaining a single high-resolution image given a set of low resolution images which are related by small displacements. We pose super-resolution as a reconstruction problem using the MRF-MAP formalism [1], and use approximate optimization using graph cuts [2] to carry out the reconstruction. We also use the same formalism to investigate high resolution expansions from single images by deconvolution assuming that the point spread function is known. We present a method for the estimation of the point spread function (PSF) for a given camera.

There have been several different approaches to super-resolution, with estimation of high-resolution (HR) images from multiple low resolution (LR) observations related by small motions being by far the most common one. Most of these methods are based on accurate registration and solve the super resolution reconstruction using variants of gradient descent with or without a smoothness prior [3, 4, 5]. Super-resolution has also been tried from multiple defocused images [6], varying zoom [7] and photometric cues [8]. Reconstruction based approaches to super-resolution model the low resolution image formation process to establish a relation between the unknown high resolution image and the low resolution observations, and use the relationship to derive algorithms to estimate the high resolution image essentially by an inversion process [6, 7, 8, 9, 10]. The inversion process is typically ill-conditioned and it often necessitates the use of smoothness or other priors [9, 11, 12] to obtain reasonable solutions. In [13], Baker and

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 385–394, 2006.

Kanade examine the limits of such processes and derive that for most point spread functions and blur kernels the estimation process is non-invertible or ill-conditioned. Further, the number of possible solutions grow at least quadratically with the desired magnification factor. They also show that this large growth in the number of solutions makes super-resolution difficult even with smoothness priors and the resulting solutions often fail to recover the high frequency details. In [14] the authors attempt to derive exact bounds on magnification factors based on a perturbation analysis. Their results indicate that under practical situations the magnification bound is only 1.6 for effective super-resolution. These results are obtained under the assumption of box PSF and local translations.

A large number of super-resolution algorithms have been based on the MAP-MRF formulation [6, 7, 8, 15] which indeed is a powerful framework for modeling the super-resolution problem. However, traditional algorithms for obtaining the MAP estimate, which in most cases result in non-convex optimization problems, have been based on simulated annealing or Iterated Conditional Mode (ICM) which provide no guarantee on the quality of the solution. Recently, Boykov et al. [2] have proposed a new algorithm based on graph-cut optimization which, under mild conditions on the nature of the objective function, can provide such guarantees. In this paper, we investigate whether with use of suitable priors, obtaining a good solution near the global optimum using an MRF-MAP formulation can indeed provide acceptable quality of reconstruction even beyond the derived limits. Unlike some methods in the literature [10, 11, 12] we do not learn the prior from examples of high resolution images, because such priors can then only be used to reconstruct similar high-resolution images. Instead, we use generic smoothness priors which are suitable for most situations.

The main contributions of this paper are as follows:

1. We give a formulation of super-resolution reconstruction from multiple displaced images using the MRF-MAP framework and solve using a graph-cut optimization. Typical graph-cut applications [16, 17] assume that the data term is a function of a single pixel. However, in the case of super-resolution, the intensity observed at a pixel is affected by neighboring pixels through a convolution representing blurring with the PSF. We formulate how such convolution based data terms can be approximated in the graph-cut formalism so that the resulting model of neighborhood interaction is regular which is necessary for graph-cut optimization [2]. We also present a method of estimation of the point spread function (PSF) of a camera as an off-line calibration process. We model the combined effects of the lens and the sensor and assume that all images are obtained using a camera whose PSF model is available.
2. We use the same formulation to deal with high-resolution expansion of single images using deconvolution.
3. Our results demonstrate that it is possible to obtain super-resolution preserving high frequency details well beyond the predicted limits of magnification of 1.6 in [14]. Note, that the limits in [14] are derived assuming box PSF and local translations and our conditions are more generic. Our results demon-

strate that even in presence of noise the super-resolved reconstruction is close to ground truth. In fact, in case of black and white images containing printed characters, we obtain high quality super-resolved images even without using a smoothness prior.

Section 2 describes the image formation process, modeling of the high resolution image as MRF. This section also gives MRF-MAP solution using graph-cut optimization for both single image expansion and SR reconstruction using multiple images. In Section 3 we present results to demonstrate the effectiveness of our method. The conclusions are given in Section 4.

2 Super-Resolution with MRF-MAP

2.1 Image Observation Model

The image observation model is given by Equation 1, as in [3, 18]

$$\mathbf{g_k} = DH_kT_k\mathbf{f} + \eta_k \qquad 1 \leq k \leq n \tag{1}$$

where $\mathbf{f}$ is the HR image, $\mathbf{g_k}$ is the k^{th} observed LR image, D is the sub-sampling matrix, T_k is the affine transformation that maps the HR image to k^{th} LR image, H_k is the space invariant PSF of the camera for the k^{th} LR image and η_k is the observation noise.

Figure 1 summarizes the observation model.

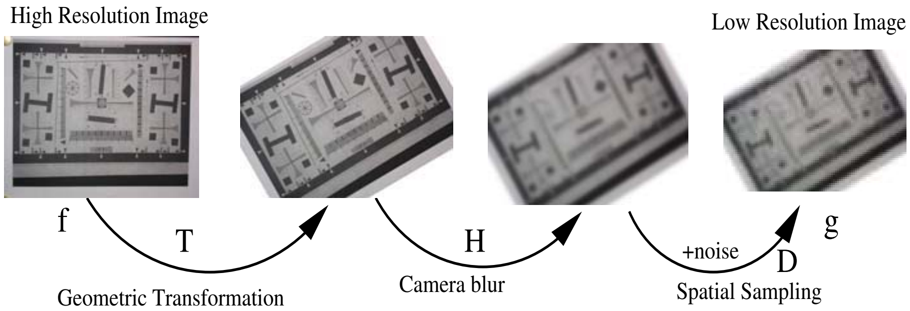


Fig. 1. Low Resolution Image Observation Model

Given the LR image and a magnification factor, the decimation matrix D is fixed. T_k can be estimated using any image registration technique, we use Hierarchical Model based Motion Estimation by Bergen et al.[19]. The estimation of PSF (H) is discussed in Section 2.2. We model the HR image as an MRF and use a maximum *a posteriori* (MAP) estimate as the final solution. The problem of SR reconstruction can be posed as a labeling problem where each pixel is assigned a label.

In the case of SR reconstruction, the posterior energy is given by

$$E(f|g) = \sum_k \|DH_kT_k\mathbf{f} - \mathbf{g_k}\|^2 + \sum_{p,q\in\mathcal{N}} V_{p,q}(f_p, f_q) \tag{2}$$

where, $V_{p,q}(f_p, f_q)$ are clique potentials which act as a smoothness prior and $\mathcal{N}$ is a neighborhood system. We minimize the posterior energy using the graph-cut technique proposed by Boykov et al. [2].

2.2 PSF Estimation

Assuming no motion blur, the edge spread function (ESF) captures the blurring effect of an ideal step edge by the image formation process. This includes both blurring due to the lens and the camera sensor. Under the Gaussian PSF assumption, the ESF $s(x)$ for a normalized edge is given by

$$s(x) = \frac{1}{2}(1 + erf(\frac{x}{\sigma\sqrt{2}})) \tag{3}$$

Given a calibration pattern with a set of ideal step edges, we estimate parameters of ESF by fitting the Equation 3 to a normalized edge in the least square sense. Note that in our MRF formalism we do not require shift invariant PSF, however, our estimates with a modern digital camera (Olympus, C-4000 zoom) indicate that the PSF is approximately space invariant. Typically the estimates of σ are around 0.4, hence we approximate the Gaussian PSF with a 3×3 mask.

2.3 Energy Minimization Using Graph-Cuts

The typical energy functions using MRF formulation are of the following form:

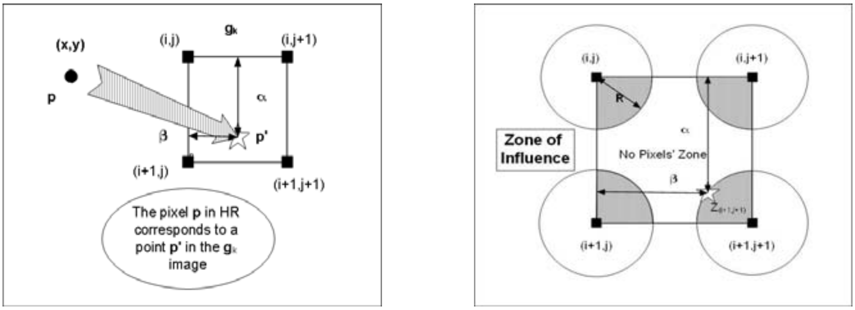
$$E\left(f\right) = \sum_{p\in\mathcal{S}} Data_p\left(f_p\right) + \sum_{p,q\in\mathcal{N}} V_{p,q}\left(f_p, f_q\right) \tag{4}$$

$Data_p\left(f_p\right)$ is a function derived from the observed data that measures the cost of assigning the label f_p to the pixel p. $V_{p,q}\left(f_p, f_q\right)$ measures the cost of assigning labels f_p, f_q to adjacent pixels p, q and is used to impose spatial smoothness. In order to minimize E using graph cuts a specialized graph is created. The form of the graph depends on the exact form of V and on the number of sites. The minimum cut on the graph minimizes the energy E either locally or globally [2].

Graph cuts can minimize only *graph-representable* energy functions. An energy function is graph-representable iff each term $V_{p,q}$ satisfies the regularity constraint [20].

For our problem the MRF sites are pixels of the HR image and labels are possible intensity values. The energy function presented in Equation 2 is in the coordinates of the LR images. In what follows, we re-write energy function in the coordinate system of the HR image.

The label at any site p is influenced by its neighbor due to the blur convolution. For a particular image site p, in the HR image, mapping to site p' in some LR



(a) (b)

Fig. 2. (a) Mapping of the Pixel from HR grid to LR grid (b) Circle of Influence of the pixel in LR grid

image potentially influences $four$ pixels $(i, j), (i, j + 1)(i + 1, j)(i + 1, j + 1)$ as shown in Fig 2(a).

We divide the space bounded by the $four$ pixels into $five$ zones – one each for the $four$ LR pixels and one *No Pixels' Zone* as shown in Fig. 2(b). The zone of each pixel is called *Zone of Influence* of that pixel. If the site p maps into one of the *Zone of Influence* of a pixel in the LR image then the expected label at the site p must be the label of that pixel. A site p mapping to a *No Pixels' Zone* of an LR image indicates that the LR image does not have any useful information about the expected label at the site p. A site p mapping to *No Pixels' Zone* for all LR images indicates that the sub-pixel displacement for this site p is not available in any of the LR images. For such a site p, we estimate the expected value of the label by interpolating the result from all the $four$ pixels in the reference LR image. From the above observation, we see that the data term for site p may be given as:

$$Data_p(f_p) = \sum_{k=1}^{n} \beta_k ||h_p * f_p - g^k_{DT_k p}||^2 \qquad (5)$$

where h_p is the blur kernel at site p and $g^k_{DT_k p}$ is the expected label at the site p with precision β_k for the k^{th} LR image. $\beta_k = 0$ for a site p that is mapped into *No Pixel Zone.* If $\beta_k = 0$ for all LR images then $\beta_1 = 1$ and $g^1_{DT_1 p}$ is set to the interpolated value from the reference LR image. Further, $\sum_{k=0}^{n} \beta_k = 1$.

The energy function is still not in the standard form for energy minimization using graph-cuts. The data term of site p also depends on the neighbors of p due to blurring operator. We now approximate the data term as a sum of two terms – (i) a term that depends only on the observed data at site p (ii) a term that depends on the neighbors of p. In following equations we assume h_p is a blur kernel with values w_{pp} at the center and w_{pq} at the neighbor q of p. Expanding we obtain the following:

$$
\begin{aligned}
Data_p(f_p) = D^*_p(f_p) + \sum_{k=1}^{n} \beta_k [\sum_{q \in \mathcal{N}_p} (2(w_{pp} f_p - g^k_{DT_k p}) w_{pq} f_q \\
+ (w_{pq} f_q)^2)] + 2 \sum_{q,r \in \mathcal{N}_p} w_{pq} w_{pr} f_q f_p
\end{aligned}
$$

$$
where, \qquad D^*_p(f_p) = \sum_{k=1}^{n} \beta_k [(w_{pp} f_p - g^k_{DT_k p})^2] \tag{6}
$$

Hence, the energy is

$$
\begin{aligned}
E(f) \approx \sum_{p \in \mathcal{C}} D^*_p(f_p) + \sum_{p,q \in \mathcal{N}} \phi_{p,q}(f_p, f_q) + \\
\sum_{p,q,r \in \mathcal{N}} \psi_{p,q,r}(f_p, f_q, f_r)
\end{aligned} \tag{7}
$$

where ϕ_{pq} is the interaction associated with two neighboring pixels and ψ_{pqr} is with three neighboring pixels, which is ignored due to first-order neighborhood approximation. For energy E to be *graph representable* the function ϕ_{pq} must be regular. But due to the term $f_p f_q$ in ϕ_{pq} the regularity condition breaks. We eliminate this dependency by further approximating $(w_{pp} f_p - g^k_{DT_k p}) = \Delta^k_p$ and $(w_{qq} f_q - g^k_{DT_k q}) = \Delta^k_q$, where Δ^k_p and Δ^k_q are fixed for a particular α-expansion move during the minimization using graph-cuts.

The equations for ϕ_{pq} after approximation is:

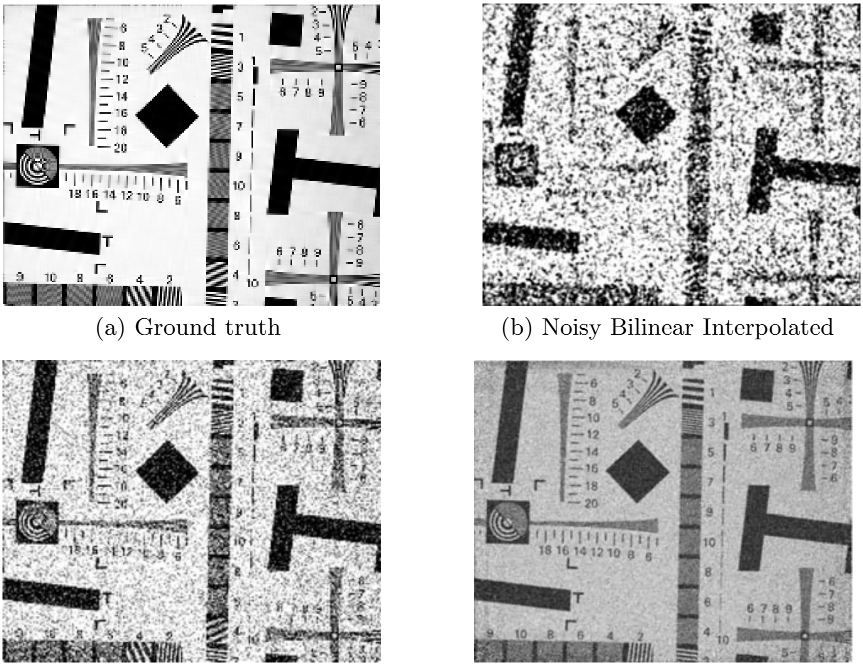
$$
\begin{aligned}
\phi_{pq}(f_p, f_q) = \sum_{k=1}^{n} \beta_k [2 \Delta_p w_{pq} f_q + (w_{pq} f_q)^2] + \\
\sum_{k=1}^{n} \beta_k [2 \Delta_q w_{qp} f_p + (w_{qp} f_p)^2] + V_{pq}(f_p, f_q)
\end{aligned} \tag{8}
$$

The single image expansion is the special case with $n = 1$ and T_1 as the identity transformation. We have used the graph-cut library provided by Kolmogrov [21] for our implementation.

3 Results

In this section we present SR reconstruction results on both synthetic and real images. In all cases the attempted magnification is $4 \times pixel - zoom$ (four in each dimension). We compare our results of single image expansion and multiple image SR reconstruction using graph-cut with bilinear interpolation and iterative back projection (IBP) method proposed by Peleg and Irani [3] (Since it is a popular method and we have its implementation). All our real images are obtained with an Olympus digital camera (C-4000 zoom).

In Figure 3, we show a synthetic example of SR reconstruction for noisy observations with a calibration image. For both SR reconstruction using multiple



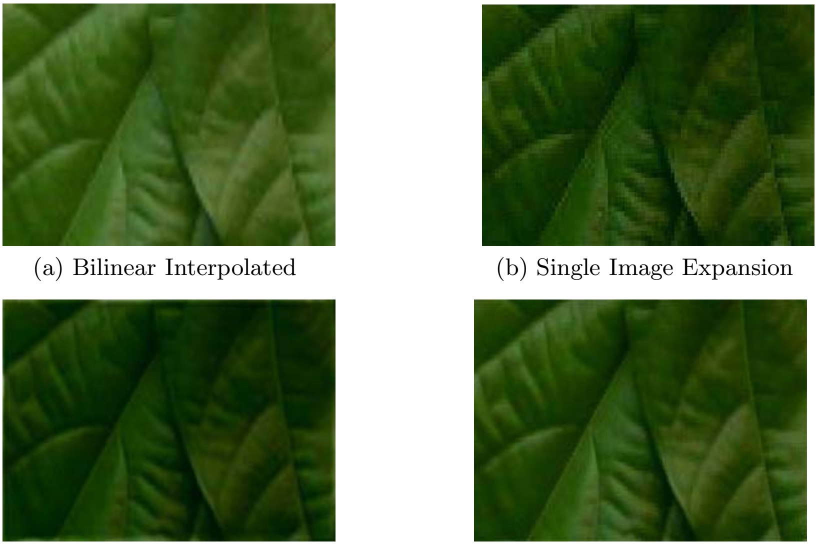
(a) Ground truth

(b) Noisy Bilinear Interpolated

(c) Single Image Expansion

(d) SR method: Multiple Images

Fig. 3. Effect of Noise on HR Image



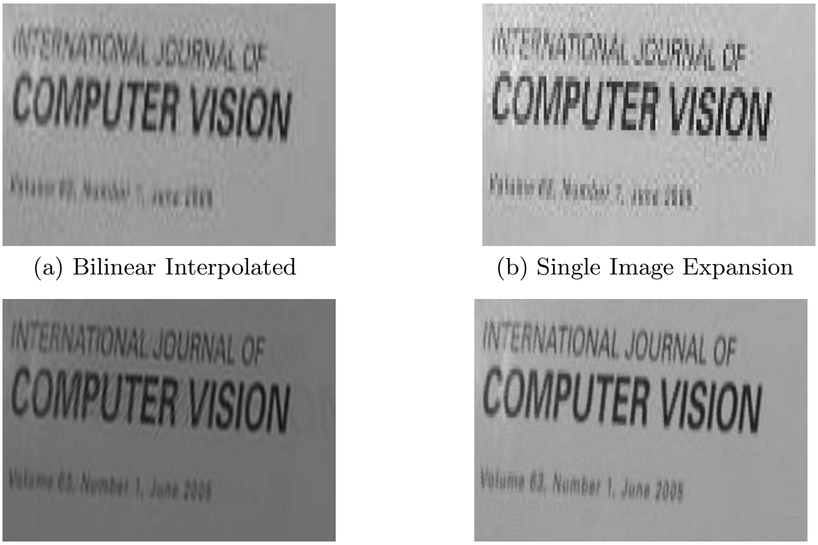
(a) Bilinear Interpolated

(b) Single Image Expansion

(c) IBP Method

(d) SR Method: Multiple Images

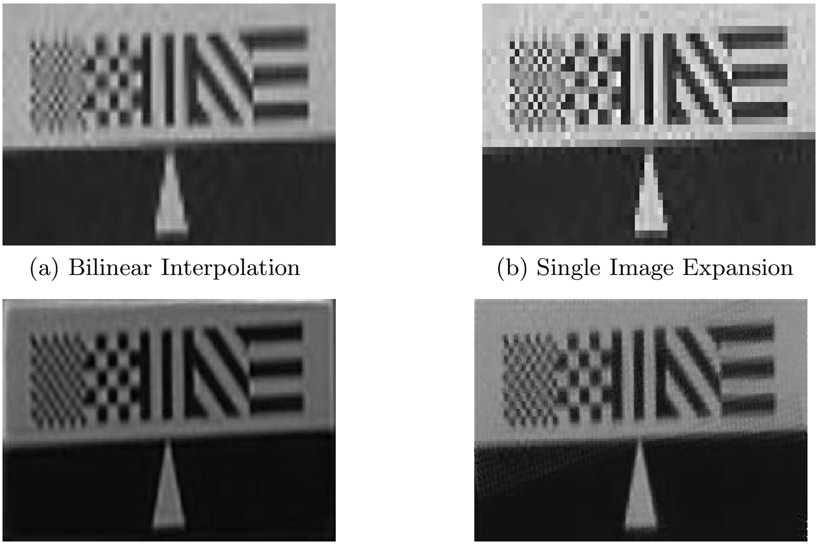
Fig. 4. HR Image of leaves with $\lambda = 0.06$



(a) Bilinear Interpolated (b) Single Image Expansion

(c) IBP Method (d) SR Method: Multiple Images

Fig. 5. HR Image of Text



(a) Bilinear Interpolation (b) Single Image Expansion

(c) IBP Method (d) SR Method: Multiple Images

Fig. 6. HR Image of Pattern

images and single image expansion we have used the linear truncated smoothness prior, given by $V_{p,q} = \lambda(min(8, |f_p - f_q|))$. We generate input images with affine transformation with additive uniform noise of SNR=2 for multiple SR reconstruction. We register 24 LR images using Hierarchical Model based Motion

Estimation by Bergen et al.[19] and carry out SR reconstruction as described in 2.3. Note that with multiple observation images the restoration method can effectively remove the noise and yet preserve the high frequency details. The results are more smooth because of registration errors in the presence of noise. Even the single image expansion can handle noise to a certain extent and even the resolution is better in the boxes in the lower left corner.

In Figures 4, 5 and 6 we show SR reconstruction using multiple images and single image expansion results for some real images. In each case for the single image expansion we use the same smoothness prior as above. For SR reconstruction using multiple images we use the same smoothness prior for results in Figure 4(d). For results in Figures 5(d) and 6(d) we do not use any smoothness prior. In each of these cases we use the same estimate of σ =0.473, since $f - number$ was fixed at 2.8.

It is evident from the results that the super resolution reconstruction using MAP-MRF using graph-cuts, both for multiple images and single image expansion, preserves the high frequency details.

The results can be downloaded from our site at http://www.cse.iitd.ac.in/˜uma/publication.html.

4 Conclusions

We have formulated the SR reconstruction problem in the framework of MRF-MAP and have proposed a solution using graph-cuts. We also carry out single image expansion using the same framework. The results demonstrate that the proposed framework for SR reconstruction using multiple images preserves the high frequency details.

The results may improve if we estimate registration parameters in the same graph-cut formulation.

Our method is not real time. We are currently exploring ways to make it fast.

References

1. S.Geman, Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Trans. on Pattern Analysis and Machine Intelligence **6** (1984) 721–741
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on PAMI **23** (2001) 1222–1239
3. M.Irani, Peleg, S.: Improving resolution by image registration. CVGIP:Graphical Models and Image Processing **53** (1991) 231–239
4. M.Irani, Peleg, S.: Motion analysis for image enhancement: Resolution, occlusion, and transparency,. Journal of Visual Communication and Image Representation **4** (1993) 324–335
5. Borman, S., Stevenson, R.: Linear models for multi-frame super-resolution restoration under non-affine registration and spatially varying PSF. In Bouman, C., Miller, E., eds.: Computational Imaging II. Volume 5299 of Proceedings of the SPIE., San Jose, CA, USA (2004) 234–245

6. D.Rajan, Chaudhuri, S.: Simultaneous estimation of super-resolved scene and depth map for low resolution defocused observations. IEEE Trans. on Pattern Analysis and Machine Intelligence **25** (2003) 1102–1117
7. Joshi, M., Chaudhuri, S.: A learning based method for image super-resolution from zoomed observations. in Proc. fifth Int. Conf. on Advances in Pattern Recognition (2003) 179–182
8. Chaudhuri, S., Joshi, M.V.: Motion-Free Super-Resolution. Springer (2004)
9. Schultz, R.R., Stevenson, R.L.: A bayesian approach to image expansion for improved definition. IEEE transctions on Image processing **3** (1994) 233–242
10. Capel, D., Zisserman, A.: Computer vision applied to super resolution. IEEE Signal Processing Magazine (2003) 75–86
11. Pickup, L.C., Roberts, S.J., Zisserman, A.: A sampled texture prior for image super-resolution. Advances in Neural Information Processing Systems (2003)
12. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE computer Graphics and Applications (2002) 56–65
13. Baker, S., Kanade, T.: Limits on super-resolution and how to break them. in Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (2000)
14. Lin, Z., Shum, H.Y.: Fundamental limits of reconstruction-based superresolution algorithms under local translation. IEEE Trans. on Pattern Analysis and Machine Intelligence **26** (2004) 83–97
15. Borman, S., Stevenson, R.: Super-resolution from image sequences – A review. In: Proceedings of the 1998 Midwest Symposium on Circuits and Systems, Notre Dame, IN, USA (1998) 374–378
16. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on PAMI **26** (2004) 1124–1137
17. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. European Conference on Computer Vision (2002)
18. Elad, M., Feuer, A.: Super resolution restoration of an image sequence: Adaptive filtering approach. IEEE Trans. on Image Processing **8** (1999) 387–395
19. Bergen, J.R., Anandan, P., Hanna, K.J., Hingorani, R.: Hierarchical model-based motion estimation. ECCV (1992) 237–252
20. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Transactions on PAMI **26** (2004)
21. Kolmogorov, V.: http://research.microsoft.com/$\sim$ vnk/. (2000)

A Multiphase Level Set Based Segmentation Framework with Pose Invariant Shape Priors

Michael Fussenegger[1], Rachid Deriche[2], and Axel Pinz[1]

[1] Institute of Electrical Measurement and Measurement Signal Processing,
Graz University of Technology,
Schießstattgasse 14b, 8010 Graz, Austria
{Fussenegger, Axel.Pinz}@tugraz.at
http://www.emt.tugraz.at

[2] INRIA, 2004 route des Lucioles, BP 93,
06902 Sophia Antipolis, France
Rachid.Deriche@sophia.inria.fr
http://www-sop.inria.fr/odyssee/

Abstract. Level set based segmentation has been used with and without shape priors, to approach difficult segmentation problems in several application areas. This paper addresses two limitations of the classical level set based segmentation approaches: They usually deliver just two regions - one foreground and one background region, and if some prior information is available, they are able to take into account just one prior but not more. In these cases, one object of interest is reconstructed but other possible objects of interest and unfamiliar image structures are suppressed.

The approach we propose in this paper can simultaneously handle an arbitrary number of regions and competing shape priors. Adding to that, it allows the integration of numerous pose invariant shape priors, while segmenting both known and unknown objects. Unfamiliar image structures are considered as separate regions. We use a region splitting to obtain the number of regions and the initialization of the required level set functions. In a second step, the energy of these level set functions is robustly minimized and similar regions are merged in a last step. All these steps are considering given shape priors. Experimental results demonstrate the method for arbitrary numbers of regions and competing shape priors.

1 Introduction

Segmenting an image into its semantically significant components is one of the fundamental problems in computer vision. Standard segmentation approaches are driven by low-level cues such as intensity, color or texture. But very often this segmentation of given objects is an ill-posed problem, therefore these methods have to fail. To overcome this limitation, prior knowledge can be used to constrain the segmentation process. Modelling this interaction between the data-driven and the model-based process has become an important topic in the research on image segmentation in the field of computer vision.

The integration of prior knowledge (in our case shape priors) into PDE based segmentation methods has delivered promising results (see [1–7]). Normally, the

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 395–404, 2006.

knowledge of one single shape prior is introduced into the contour evolution in a way that corrupted versions of a familiar object are reconstructed and all unfamiliar image structures are suppressed and often the localization of the shape must be known. Leventon et al. [3] use a Gaussian model to describe their shape priors. They assume a uniform distribution over pose parameters, that include translation and rotation. Rousson and Paragios [4] propose a similarity transformation (scale, rotation and translation) for the shape prior that allows to segment familiar objects with an unknown position in the image scene. But like the approach of Leventon et al. they can handle only one shape prior and unfamiliar image structures are ignored. Cremers et al. ([8], [6]) presented an approach with dynamic labeling, that allows to use more than one shape prior and does not suppress unfamiliar image structures. The problems of this approach are on one side the segmentation in only two regions and on the other side the incorrect segmentation of foreground objects, when one or more objects are very similar to the background. Lately, Raviv et al. [7] present a novel approach that allows a projective transformation of the shape prior, but their approach is also limited to one region. In all these approaches, it is nearly impossible to obtain the number or shapes of the unfamiliar objects in the scene. One possible way to solve that problem is to expand the level set based segmentation to an approach that allows to segment more than two regions.

For more than two regions, the level set idea loses part of its attractiveness. Therefore, there is only little related work on this problem. Paragios and Deriche [9] avoid this assumption by calculating the means of a Gaussian mixture estimation of the image histogram. The number of mixture coefficients determines the number of regions for the segmentation. Chan et al. [10] use a multiphase level set approach to segment many objects (N level-sets are used to intrinsically segment up to 2^N regions). This is a complementary approach to the one advocated in [11] to segment many objects with one level-set assigned to each object with a constraint to prevent the development of overlapping regions and/or vacuums. Brox and Weickert [12] propose a three step split and merge approach. In a first step, they use normal level set based segmentation to split the regions of an image in a recursive way. These regions are used as initialization for a level set based minimization scheme for the variational segmentation model of Zhu and Yuille [13]. In the last step, similar regions are merged to minimize the energy. All these approaches can segment different numbers of significant objects in an image, but do not use prior knowledge. In this paper, we combine the idea of level set based segmentation for multiple regions [12] with the prior knowledge of shapes to a framework which can handle these problems.

The outline of the paper is as follows: Section 2 shows a level set formulation that can easily be extended with a single shape prior. In section 3, we enhance this prior by explicit pose parameters and demonstrate the effect of a simultaneous pose optimization. In section 4, we introduce a multi region segmentation method similar to [12], that is extended with shape prior knowledge. It can handle an arbitrary number of known and unknown objects, which is also the central contribution of this work. We demonstrate that our approach is capable of reconstructing corrupted versions of multiple known objects in a scene containing other unknown objects.

2 Two Region Segmentation with a Shape Prior

There are different level set formulations, which could be possible choices [14–17]. In this work, we use the level set formulation proposed by Paragios and Deriche [17, 18] to minimize the energy for an object region:

$$E_D(\Phi, p_1, p_2) = -\int_\Omega (H(\Phi)\log p_1 + (1-H(\Phi))\log p_2)d\mathbf{x} + \nu\int_\Omega |\nabla H(\Phi)|d\mathbf{x}, \quad (1)$$

with the level set function $\Phi : \Omega \to \mathbb{R}$ with $\Phi(\mathbf{x}) > 0$ if $\mathbf{x} \in \Omega_1$ and $\Phi(\mathbf{x}) < 0$ if $\mathbf{x} \in \Omega_2$ and the Heaviside function $H(\Phi)$ with $\lim_{\Phi\to-\infty} H(\Phi) = 0$, $\lim_{\Phi\to\infty} H(\Phi) = 1$ and $H(0) = 0.5$. p_1 and p_2 are the probability densities $p_i = p(\mathbf{x}|\Omega_i)$ of the regions Ω_1 and Ω_2 which cover the whole image domain Ω with no overlap. For color images, we use the following multivariate Gaussian density:

$$p(\mathbf{x}|\Omega_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|} e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)}, \quad (2)$$

with the mean μ_i and the covariance matrix Σ_i of the multivariate color distribution of the region Ω_i. The last term of equation 1 takes into account the length of the contour weighted by the parameter ν. To add an isotropic Gaussian shape prior to the energy equation 1, we define a straight forward extension

$$E(\Phi, \Phi_0, p_1, p_2) = E_D(\Phi, p_1, p_2) + \lambda E_S(\Phi, \Phi_0), \quad (3)$$

with

$$E_S(\Phi, \Phi_0) = \int_\Omega (\Phi - \Phi_0)^2 d\mathbf{x}, \quad (4)$$

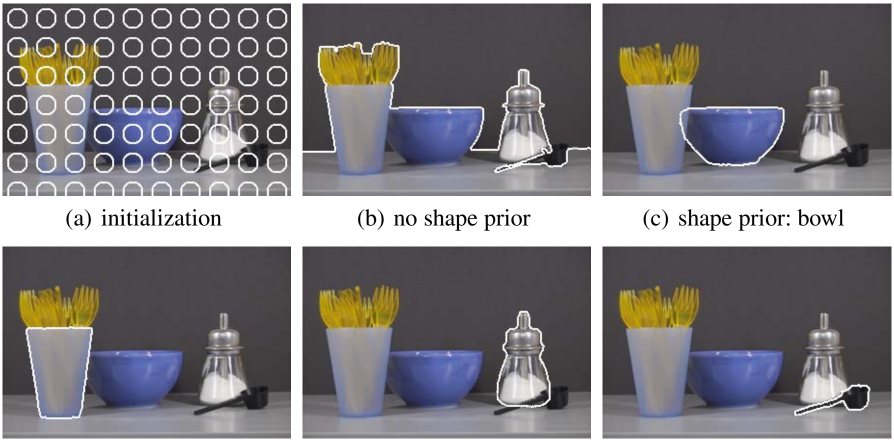
where Φ_0 is the level set of the given training shape or the mean of a set of training shapes. $\lambda \geq 0$ indicates the weight of the prior. Typically, λ is set to a value between 0.5 and 2.0.

The minimization of the energy term can now be estimated according to the gradient descent equation

$$\frac{\partial\Phi}{\partial t} = \delta(\Phi)\Big[\nu div\Big(\frac{\nabla\Phi}{|\nabla\Phi|}\Big) - log\frac{p_1}{p_2}\Big] - 2\lambda(\Phi - \Phi_0), \quad (5)$$

where $\delta(\Phi)$ is the derivative of $H(\Phi)$ with respect to its argument. The probability densities p_1 and p_2 are estimated with equation 2.

Figure 1 shows the different results of the level set segmentation with and without a shape prior. If a shape prior is used, the pose and position of the object of interest is assumed to be known. We show the original image containing different objects and the background 1(a), and the result of the standard level set segmentation without a shape prior 1(b). Subsequently, we present four results with different shape priors. With a high weight on the shape prior, the region of the specified object is correctly segmented, even if the object is partly occluded (see 1(c), 1(e)). All other objects, which are not in accordance with the given shape are suppressed. This problem is solved in section 4, but first we introduce a pose invariant formulation for the shape prior.



(a) initialization (b) no shape prior (c) shape prior: bowl

(d) shape prior: cup (e) shape prior: sugar sprinkler (f) shape prior: spoon

Fig. 1. (a) Original image with the level set initialization. (b) Level set segmentation result without prior knowledge. (c) With a shape prior of a bowl, (d) a cup, (e) a sugar sprinkler and (f) with a shape prior of a measuring spoon. With the prior knowledge of the shape, the region of its corresponding object is segmented correctly. However, it is not possible to segment the regions of two or more objects in one image, with one single level set function.

3 A Pose-Invariant Formulation

In the results shown in Figure 1, the pose and position of the object of interest is assumed to be known, but that will not be the case in realistic segmentation problems. If the object of interest is no longer presented at the same location, with the same scale and orientation as the shape prior Φ_0, a segmentation with the formalism of section 2 has to fail. Possible solutions are presented in [4], [5] and [7], where a set of pose parameters is associated with the given prior Φ_0. In our approach, we use the work of [4]. Compared to [7] the work of [4] is limited to similarity transformation, but tests have shown that on more complex scenes with more regions it is much more robust and therefore better for our use.

Rousson and Paragios [4] assume a global deformation $\mathcal{A}$ between Φ and Φ_0 that involves the parameters $[\mathcal{A} = (s; \theta; \mathbf{T})]$ with a scale factor s, a rotation angle θ and a translation vector $\mathbf{T}$. The corresponding shape energy

$$E_S(\Phi, \Phi_0(\mathcal{A})) = \int_\Omega \delta(\Phi)(s\Phi - \Phi_0(\mathcal{A}))^2 d\mathbf{x} \tag{6}$$

is simultaneously optimized with respect to the segmentation level set function Φ and the pose parameters s, θ and $\mathbf{T}$. The function is expanded with $\delta(\Phi)$, so that the shape prior is only estimated within the vicinity of the zero-crossing of the level set representation, which has a better performance than considering the whole image domain.

Minimizing equation 6 leads to the following gradient descent for the level set function Φ:

$$\frac{\partial \Phi}{\partial t} = \delta(\Phi)\Big[\nu div\Big(\frac{\nabla \Phi}{|\nabla \Phi|}\Big) - log\frac{p_1}{p_2} - 2\lambda(s\Phi - \Phi_0(\mathcal{A}))\Big]. \tag{7}$$

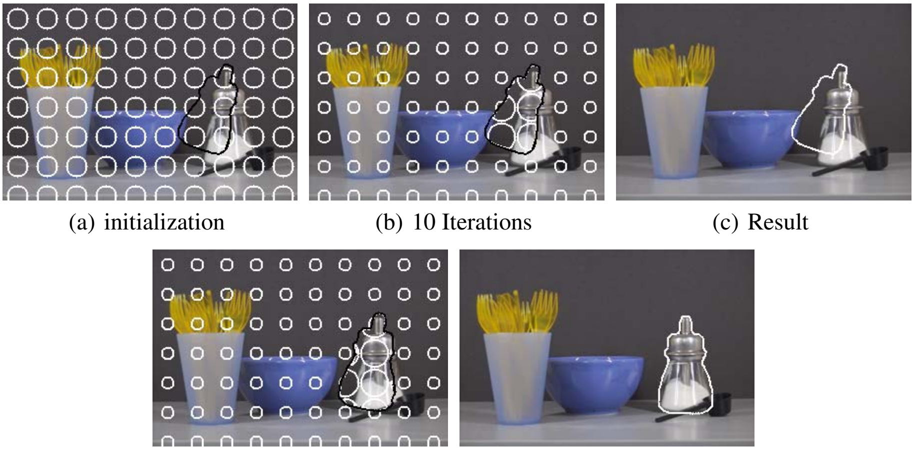
The transformation $\mathcal{A}$ is also dynamically updated to map Φ and Φ_0 in the best possible way. The calculus of variations for the parameters of $\mathcal{A}$ leads to the system:

$$\begin{aligned}
\frac{\partial s}{\partial t} &= 2\int_{\Omega} p * (-\Phi + \nabla\Phi_0(\mathcal{A}) * \frac{\partial}{\partial s}\mathcal{A})d\mathbf{x}\\
\frac{\partial \theta}{\partial t} &= 2\int_{\Omega} p * (\nabla\Phi_0(\mathcal{A}) * \frac{\partial}{\partial \theta}\mathcal{A})d\mathbf{x}\\
\frac{\partial \mathbf{T}}{\partial t} &= 2\int_{\Omega} p * (\nabla\Phi_0(\mathcal{A}) * \frac{\partial}{\partial \mathbf{T}}\mathcal{A})d\mathbf{x},
\end{aligned} \tag{8}$$

with

$$p = \delta(\Phi)(s\Phi - \Phi_0(\mathcal{A})). \tag{9}$$

Figure 2 shows the resulting segmentation of the sugar bowl with and without the pose invariant formulation. In both cases the location of the shape prior is not identical with the location of the object. Without the pose invariant formulation (top row), the familiar shape is forced to appear in a wrong position 2(c). With the pose invariant formulation (bottom row), the shape of the familiar object and the shape prior correspond 2(e).



(a) initialization

(b) 10 Iterations

(c) Result

(d) 10 Iterations

(e) Result

Fig. 2. Evolution of the shape contour (white) considering a shape prior (black). In the first row, the familiar object is forced to appear at a wrong location, without simultaneous pose optimization (figure 2(c)). In the second row, the same initialization is used but the parameters for the pose transformation are optimized. The shape of the familiar object and the shape prior correspond.

4 Multi Region Level Set Segmentation with Shape Priors

Brox and Weickert [12] introduce a split and merge level set based method to segment multiple regions. We expand their three step approach for multi region segmentation with shape priors.

The subsequent enumeration describes the three steps in detail:

1. Step 1: Splitting
   (a) For each given shape prior a split of Ω according to equation 5 is done, where the foreground region is assigned with the used shape prior and the background region is the new Ω (see figure 4(a)).
   (b) After all shape priors have been used for one split, the last Ω is split recursively using equation 5 without a prior. The final result delivers the expected number of regions in the image and is also the initialization for step 2 (see figure 4(b)).
2. Step 2: Refinement
   (a) The energy of all regions can now be minimized in a global scope with equation 10, considering also the regions assigned to the given shape priors. In the minimizing process, it can happen, that some regions become very small or even vanish. To get rid of these regions, we use the last step (see figure 4(c)).
3. Step 3: Merging
   (a) For all region pairs, where none of the two regions is assigned to a shape prior, the merged and the split energies are calculated using equation 1. If the merged energy ($E_{Merge} = E_D(\Omega_i)$) is smaller than the split energy ($E_{Split} = E_D(\Omega_{i1}) + E_D(\Omega_{i2})$) two regions are merged (see figure 4(d)).

Figure 4 shows the results after each of the above steps with two shape priors (bowl and measuring spoon). In 4(d) (final result) all objets are segmented correctly! The splittings steps and the merging step are also shown in figure 3, where every circular node of the tree symbolizes a tried split.

For the refinement we expand the gradient descent of Brox and Weickert [12] as follows:

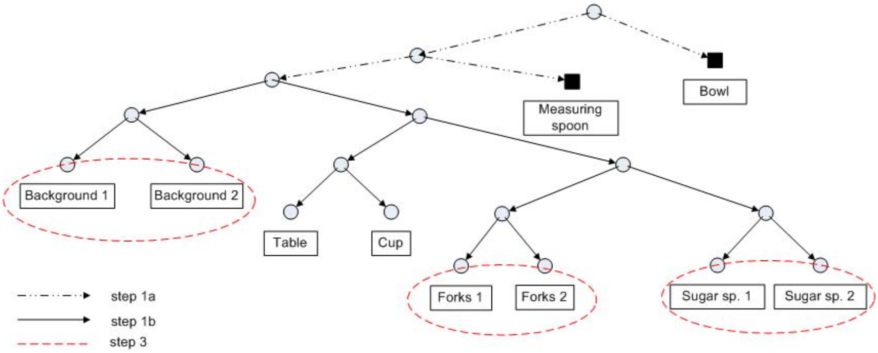
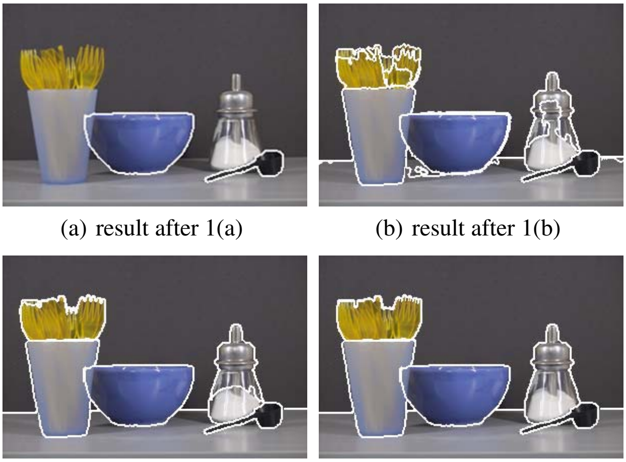


Fig. 3. Tree diagram to show the tried splittings (step 1a and 1b) and the merging (step 3)



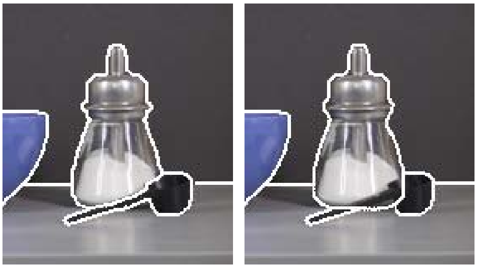
(a) result after 1(a)

(b) result after 1(b)

(c) result after 2

(d) result after 3

Fig. 4. Multi region level set segmentation with two shape priors. (a) Result of region splitting with shape priors (2 Regions), (b) after whole region splitting (10 Regions), (c) after refinement (10 Regions) and (d) final segmentation result after region merging (7 Regions). See also figure 3.



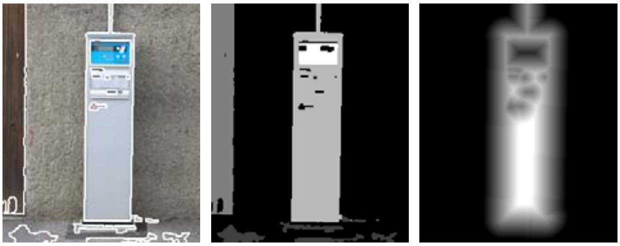
(a) $\lambda 1 = \lambda 2$

(b) $\lambda 1 > \lambda 2$

Fig. 5. Two segmentation results with varied λ for two shape priors ($\lambda 1$ sugar sprinkler, $\lambda 2$ measuring spoon)

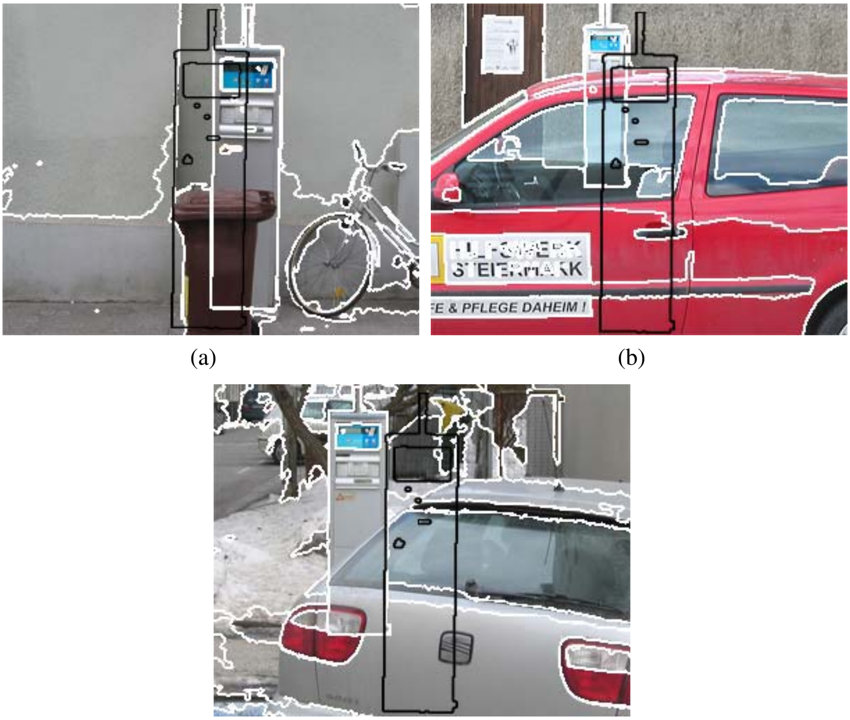
$$\frac{\partial \Phi_i}{\partial t} = \delta(\Phi_i)\Big[log p_i - \max_{j \neq i, H(\Phi_j)>0} log p_j + \frac{\nu}{2} div\Big(\frac{\nabla \Phi_i}{|\nabla \Phi_i|}\Big) \\ -2\lambda_i(s\Phi_i - \Phi_{0i}(A_i)) + 2\lambda_j(s\Phi_j - \Phi_{0j}(A_j))\Big], \tag{10}$$

where the maximum criterion ensures that a pixel is only assigned to the region with the highest probability. $\lambda_i > 0$ when Φ_i is assigned to a shape prior and $\lambda_j > 0$ when Φ_j is assigned to a shape prior, they are zero when no shape prior is assigned to the corresponding level set function. When more than one shape prior is used, it can happen that one familiar object is partially occluded by an other familiar object. If all λ have the same value the front object is segmented completely. For increasing value of λ the occluded object is fully reconstructed. That means with a variation of the different



(a) (b) (c)

Fig. 6. Multi region level set segmentation without shape prios to get a shape prior of the parking ticket machine. (a) and (b) show the segmented image. (c) shows the resulting level set function Φ_0 of the shape, that is subsequently used as shape prior for the segmentation in fig. 7.



(a) (b)

(c)

Fig. 7. Three example segmentations (white) with the initialized shape prior (black) from figure 6. The results also demonstrate the robustness of the approach. In (a) the transformation parameters $\mathcal{A}$ are $s = 0.93$, $\theta = 0.3°$ and $\mathbf{T} = [27, -2]$, in (b) $s = 0.61$, $\theta = -1.6°$ and $\mathbf{T} = [-25, -56]$ and in (c) $s = 0.82, 6$, $\theta = -1.5°$ and $\mathbf{T} = [-42, -31]$.

λ, we can give each known object different importance. Figure 5(a) demonstrates the results for an equal λ for all shape priors. In figure 5(b) the shape prior of the sugar sprinkler has a higher λ than the shape prior of the measuring spoon.

Figures 6 and 7 show an other example on real images. First in figure 6, we use the multi region level set segmentation without shape priors (figure 6(a), 6(b)) and use the segmented region of the parking ticket machine as a shape prior for the segmentations in figure 7. In all three segmentation results of figure 7(a), 7(b) and 7(c) the partly occluded parking ticket machine is segmented correctly. In all images we use small circles as the initialization for the level set function Φ and a centered level set function Φ_0 (given in black) for the shape prior. The results also illustrate the robustness of the approach.

5 Conclusion

We have introduced the framework of level set based segmentation of multiple regions, that allows to integrate an arbitrary number of competing shape priors. Each shape prior is given by a fixed template (a given training shape or the mean of a set of training shapes) and respective pose parameters. An extension to statistical shape priors, with additional deformation modes is straight forward.

First, we have shown the benefit and limitation of using a shape prior with a standard level set based segmentation. The prior knowledge permits the reconstruction of corrupted versions of a familiar object, but suppresses independent unknown objects. Furthermore, we added a pose invariant formulation.

To the end our extension to more level set functions allows us to simply use multiple competing shape priors. And additional, independent unknown objects are not suppressed. Furthermore, the different regions can be much easier distinguished and assigned to the different objects in a scene, compared to the classical approach with only one level set function. The results we have presented in this work demonstrate the power and capacity of our approach.

With its possibility to combine data-driven and recognition-driven information in the segmentation process, it can for example be used to improve an object recognition or detection framework.

Acknowledgement

This research has been partly funded by the Austrian Science Foundation (FWF, project S9103-N04), the European project IMAVIS HPMT-CT-2000-00040 within the framework of the Marie Curie Fellowships Training Sites Programme and by the European project LAVA (IST-2001-34405). This is gratefully acknowledged.

References

1. Yuille, A., Hallinan, P.: Deformable templates. In: A. Blake and A. Yuille, editors, Active Vison. (1992) 21–38
2. Cootes, T.F., A. Hill, C.J.T., Haslam, J.: Use of active shape models for locating structures in medical images. In: Image and Vison Computing. Volume 12:6. (1994) 355–365
3. Leventon, M.E., Grimson, W.E.L., Faugeras, O.: Statistical shape influence in geodesic active contour. In: Proceedings of Conference Computer Vision and Pattern Recognition. Volume 1. (2000) 316–323

4. Rousson, M., Paragios, N.: Shape priors for level set representations. In: Proceedings of European Conference of Computer Vision. Volume 2351 of LNCS. (2002) 78–92
5. Chen, Y., Tagare, H.D., Thiruvenkadam, S., Huang, F., Wilson, D., Gophinath, K.S., Briggs, R.W., Geiser, E.A.: Using prior shapes in geometric active contours in a variational framework. In: International Journal of Computer Vison. Volume 50(3). (2002) 315–328
6. Cremers, D., Sochen, N., Schnoerr, C.: Multiphase dynamic labeling for variational recognition-driven image segmentation. In: Proceedings of European Conference of Computer Vision. (2004) 74–86
7. Riklin-Raviv, T., Kiryati, N., Sochen, N.A.: Unlevel-sets: Geometry and prior-based segmentation. In: Proccedings of ECCV. (2004) 50–61
8. Cremers, D., Sochen, N., Schnoerr, C.: Towards recognition-based variational segmentation using shape priors and dynamic labeling. In: Proceedings of Scale-Space 2003. (2003) 388–400
9. Paragios, N., Deriche, R.: Coupled geodesic active regions for image segmentation: A level set approach. In: Proceedings of European Conference of Computer Vision. Volume 2. (2000) 224–240
10. Chan, T.F., Shen, J., Vese, L.: Variational PDE models in image processing. In: Notice of American Mathematical Society. Volume 50(1). (2003) 14–26
11. Zhao, H., Chan, T., Merrimann, B., Osher, S.: A variational level set approach to multiphase motion. In: Journal of Computational Physics. Volume 127. (1996) 179–195
12. Brox, T., Weickert, J.: Level set based image segmentation with multiple regions. In: Proceedings of 26th DAGM. (2004) 415–423
13. Zhu, S., Yuille, A.: Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. In: IEEE Transaction on Pattern Analysis and Machine Intelligence. Volume 18(9). (1996) 884–900
14. Osher, S.J., Sethian, J.A.: Fronts propagation with curvature depend speed: Algorithms based on Hamilton-Jacobi formulations. In: Journal of Comp. Phys. Volume 79. (1988) 12–49
15. Chan, T., Vese, L.: Active contours without edges. In: IEEE Transaction on Image Processing. Volume 10(2). (2001) 266–277
16. Tsai, A., Yezzi, A.J., Willsky, A.S.: Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation and magnification. In: IEEE Transaction on Image Processing. Volume 10(8). (2001) 1169–1186
17. Paragios, N., Deriche, R.: Geodesic active regions : a new framework to deal with frame partition problems in computer vision. In: Journal of Visual Communication and Image Representation. Volume 13(1/2). (2002) 249–269
18. Paragios, N., Deriche, R.: Geodesic active regions and level set methods for motion estimation and tracking. In: Computer Vision and Image Understanding. Volume 97(3). (2005) 259–282

A Unified Framework for Segmentation-Assisted Image Registration

Jundong Liu[1], Yang Wang[1], and Junhong Liu[2]

[1] School of Electrical Engineering and Computer Science,
Ohio University, Athens, OH, USA
liu@cs.ohiou.edu
[2] Nokia Inc., 6000 Connection Drive, Irving, TX, USA
Junhong.Liu@Nokia.com

Abstract. This paper presents a unified variational framework for seamlessly integrating prior segmentation information into non-rigid registration procedures. Under this framework, in addition to the forces arise from the similarity measure in seeking for detailed correspondence, another set of forces generated by the prior segmentation contours can provide an extra guidance in assisting the alignment process towards a more meaningful, stable and noise-tolerant procedure. Local correlation (LC) is being used as the underlying similarity measures to handle intensity variations. We present several 2D/3D examples on synthetic and real data.

1 Introduction and Related Work

Registration and segmentation are two most fundamental problems in the field of medical image analysis. Traditionally, they were treated as separate problems, each with numerous solutions proposed in literature. In recent years, the notion of integrating segmentation and registration into a unified procedure has gained great popularity, partially due to that more and more practical problems, e.g., atlas-based segmentation, subsume both segmentation and registration components.

Yezzi *et al.* [25] pointed out the interdependence existing in many segmentation and registration solutions, and a novel geometric, variational framework was then proposed that minimizes an overall energy functional involving both pre and post image regions and registration parameters. Geometrical parameters and contour positions were simultaneously updated in each iteration, and segmentations were obtained from the final contour and its transformed counterpart. While this model and its variants [18, 19] are enlightening and pave a promising way towards unifying registration and segmentation, their applicability range is either limited to relatively simple deformation type [25] (rigid/affine), or to relatively simple input images [18, 19].

Vemuri *et al.* [27] propose a segmentation + registration model to solve the atlas-based image segmentation problem where target image is segmented

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 405–414, 2006.

through the registration of the atlas to the target. A novel variational formulation was presented which put segmentation and registration processes under a unified variational framework. Optimization is achieved by solving a coupled set of nonlinear PDEs.

Another segmentation + registration model proposed by Noble *et al.* [13] seeks for the best possible segmentation and registration from the *maximum a posteriori* point of view. Improvements in accuracy and robustness for both registration and segmentation have been shown, and potential applications were identified. This model is primarily designed for combining segmentation and *rigid* registration. While non-rigid algorithm was also implemented, the motion field estimation is based on block-matching of size (7×7), which is not dense enough for most non-rigid registration applications.

Inspired by the above-mentioned approaches, the work presented in this paper is aimed to establish a segmentation assisted framework to boost the robustness of non-rigid image registration. Another component of our method, aiming to achieve the same goal, is the choosing of the *sum of Local Correlation (LC)* as the underlying similarity measure. *LC* measure is invariant to intensity scaling and contrast and this property is very crucial for registration applications where input data have substantial intensity variations. The following is a brief review on some of the related similarity measures that can handle intensity variations.

Intensity-Variation-Tolerant Measures. A group of similarity measures that don't assume *brightness constancy* are based on the concept of maximizing mutual information reported in Viola and Wells [21], Collignon *et al.*, [3] and Studholme *et al.*, [17]. One noticeable drawback of MI-based registration [9] is its proneness of being trapped in local maxima, as no spatial information has been taken into consideration in the original formulation [21, 17]. Although some studies, such as combining MI with gradient information [14] and using regional MI [5], showed impressive improvements for boosting the robustness for MI-based elastic registration, no systematic solution has been proposed up to this date.

Recently, the *sum of Local Correlation (LC)* [23, 12], derived from image statistics (mean and variance), started to show great successfulness in registering mono-modal and multi-modal images with impressive accuracy and efficiency. Due to the fact that it can capture both the statistical correspondence and the spatial coherence existing in the input images, *LC* is fairly responsive to local spatial changes, therefore has a great potential to reveal very detailed non-rigid motions. In addition, *LC* can be formulated [23] to be relatively invariant to intensity scaling and reversal, which makes it an ideal similarity metric for handling the registration problems where "subtle spatial changes" and "presence of intensity variations" are two major characteristics.

Proposed Registration Method. In this paper, we propose to develop a robust segmentation-guided registration framework, with *LC* as the underlying matching metric. Under this framework, prior shape information can be fully integrated into the registration procedure, as an extra guiding force to lead a more meaningful, stable and noise-tolerant image alignment process.

Our segmentation-guided model is inspired by the works mentioned in the previous section, and it differs from other models in that 1) our method is a fully non-rigid dense deformation estimation model; 2) it uses a unified segmentation + registration energy minimization formulation, and 3) the optimization is carried out under a natural, parameterization-free and numerically stable level set framework. **A salient feature of our model is its robustness against input image noise.**

With different similarity measure embedded into our framework, our model can handle both single-modality and multi-modality image registrations. In this paper, we propose a modified LC as the underlying similarity metric. Comparing with the existing LC formulations, our LC formula has the advantage (over [12]) of being able to handle both global and local motion, and advantage (over [23]) of being able to handle both intensity scaling and reversal.

The outline of the paper is as follows. In the next section, we introduce our segmentation guided registration model as an energy minimization and formulate it under level set framework. Associated Euler-Lagrange equations will be provided and discussed. In section 3, we give experiment results to demonstrate the performance of our algorithm on several 2D images. We conclude this paper in section 4.

2 Segmentation Guided Registration Model

Commonly, the basic input data to a registration process are two images: one is defined as *fixed* (or *target*) image $I_1(X)$ and the other as the *moving* (or *source*) image $I_2(X)$. A typical solution to the non-rigid registration problem is to look for a deformation function V assigned to each point X. The function is searched by minimizing an energy function E of the form

$$E(V) = S(V) + R(V) \tag{1}$$

The term $S(V)$ is designed to measure the dis-similarity between the input image I_1 and I_2. The term $R(V)$ is designed to penalize fast variations of the deformation function V.

In addition to these two image, our model requires a segmentation of the fixed image, indicting a studying area of $I_1(X)$, as another input component. Let C be the boundary curve of the segmentation. We denote by C_{in} and C_{out} representing the inside and outside areas of the curve C. Let C_1 and C_2 be the average values for C_{in} and C_{out} respectively.

The contour C can be either input by user or derived from a training set. We assume that the region captured by C contains a single object of the fixed image, therefore the intensity profiles of both inside and outside of the region should be able to be characterized by certain property. Examples of the property include “being relatively homogenous”, or “conforming to certain distribution”. Suppose the fixed and moving images are well corresponded, then, at the time a perfect alignment is achieved, the intensities in the warped moving image should also have a similar property within both C_{in} and C_{out}. This observation provides the justification for our model, which is designed based on following considerations:

> In addition to the set of forces generated by intensity similarity measure (e.g., SSD, *LC* or MI) to warp the moving image toward the target, another set of forces, derived from the region property constraint, should be utilized to pull the moving image toward the correct alignment. This set of forces can provide an extra guidance for the registration process to avoid local energy optima, which is especially helpful when input images are noisy.

Our solution to the segmentation guided registration can be formulated as the minimization of a new energy, which integrates the available segmentation information,

$$E(V) = S(V) + H(V) + R(V) \tag{2}$$

where $H(V)$ is designed to penalize the deformations that would result in inhomegeneous intensity profiles within C_{in} and C_{out}.

For different applications, we can assign different forms for the terms $S(V)$, $H(V)$ and $R(V)$. $S(V)$ term can take Sum of Squared Difference (SSD) for single-modality applications and Mutual Information (MI) and Correlation Ratio (CR) for multimodal registration. For $H(V)$, piecewise constant function [2] and piecewise linear function [16] are among the most popular choices. Gaussian diffusion model, elastic model and viscous fluid model have been widely used as the regularization options for term $R(V)$.

2.1 Frameworks Based on Intensity Homegeneity

To handle the intensity variations existing in the input images, our solution to the robust segmentation-guided registration is formulated as the minimization an energy, which relies on *LC* to measure the image similarity, with the available segmentation information being used as a homogeneity constraint,

$$\begin{aligned} E(V) = &\int_{\Omega} LC(I_1(X), I_2(X + V(X)))dX + \lambda_1 \int_{C_{in}} \left[I_2(X + V(X)) - C_1\right]^2 dX \\ &+ \lambda_2 \int_{C_{out}} \left[I_2(X + V(X)) - C_2\right]^2 dX + \lambda_3 \int_{\Omega} | \bigtriangledown V(X)|^2 dX \end{aligned} \tag{3}$$

where $LC(I_1, I_2)$ is the *LC* similarity between I_1 and transformed I_2. Ω is the image domain and $V(X)$ denotes the deformation field. λ_1, λ_2 and λ_3 are three constant parameters that weight the importance of each term in the optimization energy.

Several variations of *LC* measures have been investigated in Cashier *et al.* [1] and Netsch *et al.* [12]. In our implementation, we use a different customized form of local correlation and it has the advantage (over [12]) of being able to handle both global and local motion, and advantage (over [23]) of being able to handle both intensity scaling and reversal. We formulate the local correlation measure as follows,

$$LC(I_1, I_2) = \sum_b \frac{\sum\limits_{a\in n(b)} (i_{1_a} - \bar{i}_{1_b})^2 (i_{2_a} - \bar{i}_{2_b})^2}{\left[\sum\limits_{a\in n(b)} (i_{1_a} - \bar{i}_{1_b})^2 \sum\limits_{a\in n(b)} (i_{2_a} - \bar{i}_{2_b})^2\right]}$$

where $i_1 = I_1(X)$ and $i_2 = I_2(X + V(X))$ with a representing the pixels in the neighborhood $n(b)$ around pixel b in the image. In the energy function $E(V)$, the first term $LC(I_1, I_2)$ in the energy function provides the main force for matching two images, while $\int_{C_{in}} \left[I_2(X + V(X)) - C_1\right]^2 dX$ and $\int_{C_{out}} \left[I_2(X + V(X)) - C_2\right]^2 dX$ terms allow the a priori segmentation to exert its influence, aiming to enforce the homogeneity constraints. $\int_{\Omega} | \bigtriangledown V(X)|^2 dX$ is a diffusion term to smooth the deformation field.

2.2 Level Set Formulation of the *LC*-Based Model

The energy function $E(V)$ can be minimized under the level set framework. Introduce a continuous function $\phi : \Omega \to R$, so $C = \{(X) \in \Omega : \phi(X) = 0\}$, and we choose ϕ to be positive in C_{in} and negative in C_{out}. We adopt the model presented in Chan *et al.* [2] and we have the following functional:

$$\begin{aligned} E(V) = &\int_{\Omega} LC(I_1(X), I_2(X + V(X)))dX + \lambda_1 \int_{\Omega} \left[I_2(X + V(X)) - C_1\right]^2 H(\phi(X))dX \\ &+\lambda_2 \int_{\Omega} \left[I_2(X + V(X)) - C_2\right]^2 (1 - H(\phi(X)))dX + \lambda_3 \int_{\Omega} | \bigtriangledown V(X)|^2 dX \end{aligned} \qquad (4)$$

where H is the Heaviside function. The Euler-Lagrange differential equation of this functional is given by:

$$\begin{aligned} \tfrac{dE}{dV} = &\frac{d(LC)}{dV} + 2\lambda_1 (I_2(X + V) - C_1) \bigtriangledown I_2(X + V) \cdot H(\phi(X)) \\ &+2\lambda_2 (I_2(X + V) - C_2) \bigtriangledown I_2(X + V) \cdot (1 - H(\phi(X))) + \lambda_3 \bigtriangledown^2 V \end{aligned}$$

where

$$\begin{aligned} \tfrac{d(LC)}{dV} = \sum_b \frac{2}{\sum\limits_{a\in n(b)} (i_{1_a} - \bar{i}_{1_b})^2} \Bigg[&\frac{\sum\limits_{a\in n(b)} (i_{1_a} - \bar{i}_{1_b})^2 (i_{2_a} - \bar{i}_{2_b})}{\sum\limits_{a\in n(b)} (i_{2_a} - \bar{i}_{2_b})^2} \\ &- \frac{\sum\limits_{a\in n(b)} (i_{1_a} - \bar{i}_{1_b})^2 (i_{2_a} - \bar{i}_{2_b})^2 \sum\limits_{a\in n(b)} (i_{2_a} - \bar{i}_{2_b})}{\sum\limits_{a\in n(b)} (i_{2_a} - \bar{i}_{2_b})^2} \Bigg] \bigtriangledown i_2 \end{aligned}$$

and

$$C_1 = \frac{\int_\Omega I_2(X+V)H(\phi(X+V))dX}{\int_\Omega H(\phi(X+V))dxdy}$$

$$C_2 = \frac{\int_\Omega I_2(X+V)(1-H(\phi(X+V)))dX}{\int_\Omega (1-H(\phi(X+V)))dxdy}$$

To estimate the deformation field between I_1 and I_2, we initialize the deformation field as $X(V) = 0$ at each pixel, and use the Euler equation as a gradient descent process that eventually leads to the convergence of the alignment process. The level set function being used in this paper is $\phi(X, 0) = D(X)$, where $D(X)$ is the signed distance from each grid point to the zero level set C. This procedure is standard, and we refer the reader to [26] for details.

2.3 A SSD-Based Segmentation + Registration Model

For comparison purpose, we also provide the sum of squared (SSD) based segmentation-guided model, which is to minimize the following energy,

$$\begin{aligned} E(V) = \int_\Omega \left[I_1(X) - I_2(X + V(X))\right]^2 dX + \\ H(V) + R(V) \end{aligned}$$

where the homogeneity and regularization parts are identical to those in Eqn. (4).

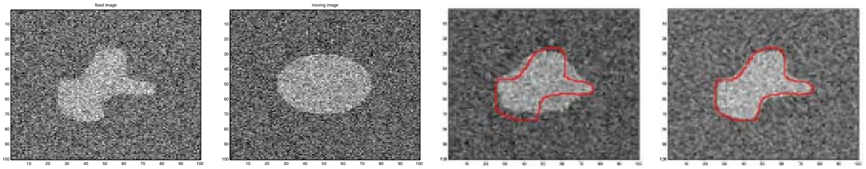
3 Experimental Results

In this section, we present two sets of experiments to demonstrate the improvement made by the two components of our proposed registration method: the segmentation-guided framework and local correlation.

3.1 Registration Based on the Segmentation + Registration Component

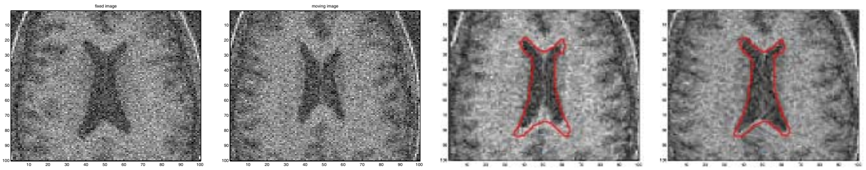
Three examples are used to test the segmentation + registration component. In all cases, we will compare the results using our model with that of using the famous *Demons* algorithm [6]. Here, *Demons* algorithm is counted as a representative of those “registration-only” approaches. In consideration that *Demons* algorithm is a SSD-based method, in order to make the comparison more meaningful, our model being used for these three examples is a *segmentation + registration + SSD* version, as formulated in section 2.3. The goal of these examples is to demonstrate the helpfulness of integrating segmentation information into the registration procedure, especially in handling image noise.

The first example contains a pair of synthetically generated images, where the fixed image was generated from the moving by a known non-rigid field. Zero-mean Gaussian noise was then added to each image. The standard deviation is 20. Fig. 1.a and 1.b show the two images. In the following examples, we chose



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

Fig. 1. Registration results for a pair of synthetic images. (a) is the fixed image and (b) the moving image. (c) is the registration result of using the *Demons* algorithm, and (d) using our segmentation guided registration model. The edge map from the fixed image is superimposed.



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

Fig. 2. Registration results for a pair of 2D MR images. For details, see text.

the constants $\lambda_1 = 0.1$, $\lambda_2 = 0.1$ and $\lambda_3 = 1$, respectively. The segmentation of the fixed image was manually obtained, as superimposed on the moving image in Fig. 1.c. Two registration approaches: the *Demons algorithm* as well as our *segmentation-guided registration* model are then applied. We should note that the *Demons* algorithm relies on intensity alone for registration.

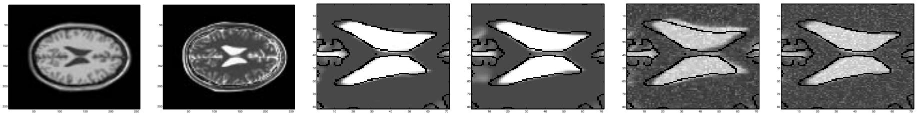
The results are also shown in Figure 1. Fig. 1.c is the transformed moving image from the *Demons* algorithm. Fig. 1.d shows the result of our model. As evident, the *Demons* algorithm had trouble in warping the moving image to a perfect matching, which is partially due to the numerous local energy minima resulted from the huge amount of noise existing in the images. However, the registration result generated from our model is quite accurate, which indicates that the integrated segmentation information is very helpful in pulling the moving image towards a correct matching.

We designed and carried out a similar experiment on a pair of MRI brain slices. The two slices have substantial disparity in shape of the ventricles, which is the region of interest. Figure 2 shows the images and results. Fig. 2.a and 2.b are the fixed and moving images respectively. Fig. 2.c and 2.d depict the results from the *Demons* algorithm (2.c) and our segmentation guided registration model (2.d). As evident, the former model fails to transform the ventricle area into a desired position, while the latter accurately achieves the registration goal.

3.2 Registrations of Our Segmentation + Registration *LC* Model

In this section, we demonstrate the performance of our Segmentation + Registration with LC as the similarity measure. In order to test the functionality of our “registration + segmentation *LC*”, we made a “registration-only *LC*” model as the comparison, which is obtained by turning off the the segmentation input (setting the weighting factor of the segmentation component to zero) in our model.

The experiment is conducted on a pair of T1/T2 brain slices. In order to demonstrate *LC*'s ability of handling multi-modality images, we set the region of interest as the area around the ventricle, where two slices have substantial disparity. Figure 3 shows the images and results. Fig. 3.a and 3.b are the fixed and moving images respectively. Fig. 3.c and 3.d depict the results from the “registration only *LC*” model (3.c) and the “registration + segmentation *LC*” model (3.d). As evident, both models can obtain fairly accurate matching for this clean image pair. To demonstrate the ability of the “segmentation + registration” component in handling noisy image data, we applied a zero-mean Gaussian noise with standard deviation of 10 onto both input images. Fig 3.(e) and 3.(f) are the results from “registration-only” and “segmentation + registration” models, respectively. It's clearly shown that, the former fails to transform the ventricle area into a desired position, while the latter accurately achieves the registration goal.



(a) (b) (c) (d) (e) (f)

Fig. 3. Registration results for a pair of T1/T2 images. First row: (a) fixed image, (b) moving image; (c) deformed moving image using the “registration only *LC*” model, (d) deformed moving image using our “registration + segmentation *LC*” model. Second row: same experiment with noisy image inputs; results from the “registration only” (e) and “registration + segmentation *LC*” model (f).

In summary, our “segmentation + registration *LC*” model has the desired property of being insensitive to intensity reversal, scaling as well as image noise, therefore it has the great potential to be used to accurately and robustly register medical images, especially when certain segmentation information is available.

4 Conclusions

In this paper, we present a segmentation-guided non-rigid registration algorithm, which integrates the available prior shape information as an extra forces to lead to a noise-tolerant registration procedure. Our model differs from other methods in that we use a unified segmentation + registration energy minimization formulation, and the optimization is carried out under level-set framework. *Local Correlation* has been used as the similarity measure to handle intensity variations. We showed the improvement made with our model by comparing the results with that of the Demons algorithm. To explore other similarity metrics under the same framework to handle more complicated inputs will be the focus of our future work.

References

1. Cachier Pascal, Pennec Xavier, "Non-Rigid Registration by Gradient Descent on a Gaussian-Windowed Similarity Measure using Convolutions", *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pp.182, 2000.
2. T. Chan, L. Vese, "An Active Contour Model without Edges", *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 266-277.
3. A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, and P. S. ang G. Marchal, "Automated multimodality image registration using information theory," *Proc. IPMI*, pp. 263-274, 1995.
4. A. Roche, G. Malandain, X. Pennec, and N. Ayache. "Multimodal Image Registration by Maximization of the Correlation Ratio". Research Report RR-3378, INRIA, August 1998.
5. Daniel B. Russakoff, Carlo Tomasi, Torsten Rohlfing, and Calvin R. Maurer, Jr, "Image Similarity Using mutual information of Regions", *ECCV 2004*, LNCS 3023, pp. 596-607, 2004.
6. J. P. Thirion, "Image matching as a diffusion process: an analogy with Maxwell's demons", *Medical Image Analysis*, Volume 2, Issue 3, September 1998, pp 243-260.
7. T. Zhang and D. Freedman, "Tracking Objects Using Density Matching and Shape Priors", *ICCV 2003*, Vol. 2, pp. 1056-1062, 2003.
8. M. Leventon and W. E. L. Grimson, "Multi-modal volume registration using joint intensity distributions," in MICCAI 1999.
9. Jundong Liu and Junhong Liu, "Artifacts Reduction in Mutual Information-based Image Registration using Prior Information", The International conference on Image Processing (ICIP 2003), Sept. 14-17, Barcelona, Spain, 2003.
10. J.B. Maintz and M. A. Viergever,"A Survey of Medical Image Registration," *MedIA* Vol. 2, pp. 1-36,1998.
11. C. Meyer et. al, "Demonstration of Accuracy and Clinical Versatitlity of Mutual Information for Automatic Multimodality Image Fusion using Affine and Thin-plate Spline-warped Geometric Deformation", *Medical Image Analysis*, vol. 1, 195-206, 1997.
12. T. Netsch et. al, "Towards Real-Time Multi-Modality 3D Medical Image Registration", pp. 718 - 725, ICCV 2001.
13. P.P. Wyatt and J.A. Noble, "MAP MRF joint segmentation and registration of medical images", *Medical Image Analysis*, vol. 7, pp. 539-552, 2003.
14. Pluim JP, Maintz JB, Viergever MA, "Image registration by maximization of combined mutual information and gradient information", *IEEE Trans Med Imaging*, Volume: 19, Issue: 8, pp 809-814, Aug 2000.
15. D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, "Non-rigid registration using Free-Form Deformations: Application to breast MR images", *IEEE Transactions on Medical Imaging*, 18(8):712 - 721, 1999.
16. A Tsai, AJ. Yezzi, and A. Willsky, "A Curve Evolution Approach to Smoothing and Segmentation Using the Mumford-Shah Functional", CVPR 2000: 1119-1124.

17. C. Studholme, D. Hill and D. Hawkes, “An overlap invariant entropy measure of 3D medical image alignment,” *Pattern Recognition*, Vol. 32, pp. 71-86,1999
18. G. Unal, G. Slabaugh, A. Yezzi and J. Tyan, “ Joint segmentation and non-rigid registration without shape priors”, Siemens Technical Report SCR-04-TR-7495, 2004.
19. G. Unal and G. Slaaugh, “Coupled PDEs for Non-Rigid Registration and Segmentation”, *CVPR 2005*, San Diego, CA, 2005.
20. B. Vemuri, J. Ye, Y. Chen, and C. Leonard, “Image registration via level-set motion: applications to atlas-based segmentation.” Medical Image Analysis, 7(1):1-20, March, 2003.
21. P. A. Viola and W. M. Wells, “Alignment by maximization of mutual information,” in *Fifth ICCV*, MIT, Cambridge, MA, pp. 16-23, 1995.
22. Y. Wang and L. H. Staib, “Physical model-based non-rigid registration incorporating statistical shape information,” *Medical Image Analysis*, vol. 4, no. 1, pp. 7-20, 2000.
23. Weese J1rgen, Rosch Peter, Netsch Thomas, Blaffert T., Quist Marcel “Gray-Value Based Registration of CT and MR Images by Maximization of Local Correlation”, *MICCAI 99*, pp. 656 - 663, 1999.
24. W. M. Wells III, P. Viola, and H. Atsumi, “Multi-modal Volume Registration by Maximization of Mutual Information,” *Medical Image Analysis*, 1(1), pp. 35-51, 1997.
25. A. Yezzi, L. Zollei and T. Kapur, “A Variational Framework for Joint Segmentation and Registration”, *Medical Image Analysis*, pp 171-185, June 2003.
26. M. Sussman, P. Smereka and S. Osher, “A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow”. *Journal of Computational Physics*, vol. 114, pp. 146–159, 1994.
27. B. Vemuri, Yun Chen and Z. Wang, “Registration Assisted Image Smoothing and Segmentation”. *ECCV* (4) pp. 546-559, 2002.

Fusion of 3D and Appearance Models for Fast Object Detection and Pose Estimation

Hesam Najafi[1], Yakup Genc[1], and Nassir Navab[2]

[1] Real-time Vision and Modeling Department, Siemens Corporate Research, Inc.,
Princeton, NJ 08540, USA
[2] Institut für Informatik, Technische Universität München,
Boltzmannstr. 3, 85748 Garching bei München, Germany
{Hesam.Najafi, Yakup.Genc}@siemens.com,
navab@cs.tum.edu

Abstract. Real-time estimation of a camera's pose relative to an object is still an open problem. The difficulty stems from the need for fast and robust detection of known objects in the scene given their 3D models, or a set of 2D images or both. This paper proposes a method that conducts a statistical analysis of the appearance of model patches from all possible viewpoints in the scene and incorporates the 3D geometry during both matching and the pose estimation processes. Thereby the appearance information from the 3D model and real images are combined with synthesized images in order to learn the variations in the multiple view feature descriptors using PCA. Furthermore, by analyzing the computed visibility distribution of each patch from different viewpoints, a reliability measure for each patch is estimated. This reliability measure is used to further constrain the classification problem. This results in a more scalable representation reducing the effect of the complexity of the 3D model on the run-time matching performance. Moreover, as required in many real-time applications this approach can yield a reliability measure for the estimated pose. Experimental results show how the pose of complex objects can be estimated efficiently from a single test image.

1 Introduction

Estimating the pose of a camera relative to an object is one of the most studied problems in computer vision and photogrammetry. While reliable solutions have been proposed for pose estimation given correspondences [1–4]and feature-based 3D tracking [5–7], fully automated estimation of the initial camera's pose for tracking is still an open problem. The difficulty stems from the need for fast and robust detection of known objects in the scene given their 3D models, or a set of 2D images or both. Fast and robust pose estimation has a wide variety of applications, such as robot navigation, surveillance, and augmented reality.

Computer vision literature includes many object detection approaches [8, 9, 5, 10, 11] based on representing objects of interests by a set of local features which are characterized by invariant descriptors for matching [12–16]. Combination of such descriptors provide robustness against partial occlusion and cluttered

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 415–426, 2006.

backgrounds. The descriptors are ideally invariant to viewpoint and illumination variations. Most of these methods make use of techniques for wide-baseline stereo matching solely based on 2D images without considering any run-time requirements. However, in many applications where real-time object detection is required both 3D models and several training images may be available or can be created easily during an off-line process.

This paper presents an alternative approach for fast object detection and pose estimation by fusing both 3D and appearance models. It shows that real-time performance can be achieved by using the underlying 3D information to limit the number of hypothesis for the robust matching process. Especially for large environments this renders our method very powerful. Our method differs in two aspects from the state of the art. First, we propose a statistical analysis and evaluation of the appearance and shape of features from all possible viewpoints in the scene combining real and synthetic viewpoints. Second, we make use of the known 3D geometry in both matching and pose estimation processes. We show that by fusing both appearance and geometric information rather than using them in separate procedures we can improve both time and functional performance, and make our approach more scalable for large environments.

Our approach has two phases. In the training phase, a compact appearance and geometric representation of the target object is built. This is as an off-line process. The second phase is an on-line process where a test image is processed for detecting the target object using the representation built in the training phase. During training, the variations in the descriptors of each feature are learned using principal component analysis (PCA). Furthermore, for each feature a reliability measure is estimated by analyzing the computed visibility distribution from different viewpoints. The problem of finding matches between sets of features in the test image and on the object model is then formulated as a classification problem which is constrained by using the reliability measure of each feature.

As an application, our method is intended to be used to provide robust initialization for a frame rate feature-based pose estimator [6] where robustness and time efficiency are very critical. In this case the initial pose recovery is sufficient to be performed under one second.

2 Previous Work

A number of approaches have been proposed addressing the problem of 3D object detection for pose estimation. Some methods use statistical classification techniques, e.g PCA to compare the test image with a set of calibrated training images [17]. Others are based on matching of local image features [12, 13, 18, 19, 5, 20, 21, 22, 23, 24]. While some approaches use simple 2D features such as corners or edges, more sophisticated approaches rely on local feature descriptors which are insensitive to viewpoint and illumination changes. Usually geometric constraints are used as verification criteria of the estimated pose. Rothganger et al. [20] introduced a 3D object modeling and recognition algorithm for affine viewing conditions. Photometrically and geometrically consistent matches are selected in a RANSAC-based pose estimation procedure. Even though this

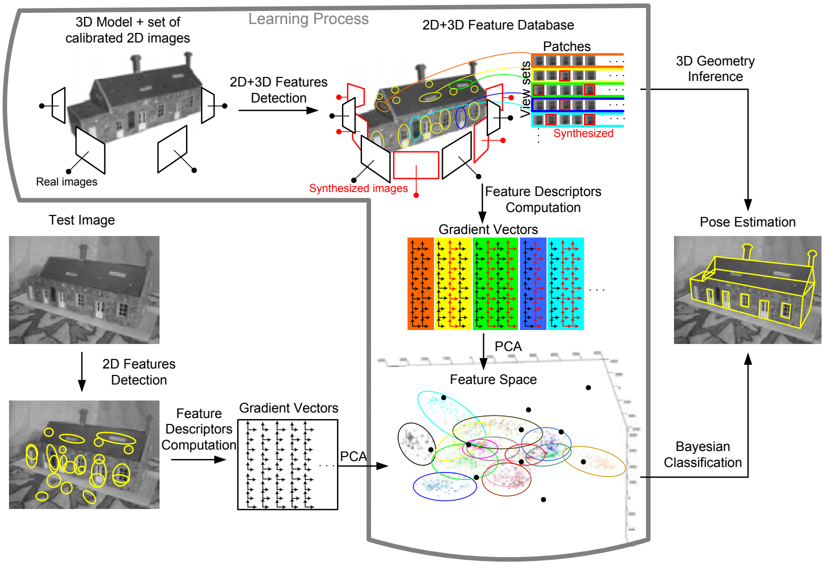


Fig. 1. Overview of the proposed object detection process for real-time pose estimation

method achieves good results for 3D object detection, it is too slow for real-time applications. Lepetit et al. [19] treat wide baseline matching of key points as a classification problem, where each class corresponds to the set of all possible views of each point. Once potential matches have been established they apply a plain RANSAC method to recover the 3D pose. Recently they introduced an approach for object pose estimation in real-time [25], where randomized trees are used as the classification technique. Keypoint recognition relies solely on 2D image intensity values within small windows around these keypoints.

3 Proposed Approach

Our goal is to automatically detect objects and recover their pose for arbitrary images (*test image*). The proposed object detection approach is based on two stages: A learning stage which is done off-line and the matching stage at run-time. The entire learning and matching processes are fully automated and unsupervised. Sections 3.1 and 3.2 describe the learning step in more detail. In Sections 3.3 and 3.4 we introduce the matching and pose estimation algorithms that enforce both photometric and geometric consistency constraints.

3.1 Creating View Sets Based on Similarity Maps

In the first step of the learning stage a set of stable feature regions are selected from the object by analyzing their detection repeatability and accuracy as well as their visibility from different viewpoints.

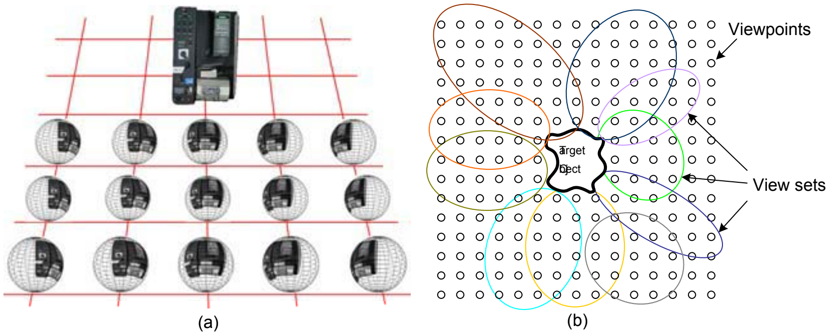


Fig. 2. (a) A subset of the environment maps surrounding the object of interest. (b) A 2D illustration of the 3D clusters of the view sets surrounding the target object.

Images represent a subset of the sampling of the so called *plenoptic function* [26]. The plenoptic function is aparameterized function for describing everything that can be seen from all possible viewpoints in the scene. In computer graphics terminology the plenoptic function describes the set of all possible *environment maps* for a given scene. In our case, we define a complete sample of the plenoptic function as a full spherical environment map (see Fig. 2(a)). Having a set of calibrated images and the virtual model of the target object, the viewing space is coarsely sampled at discrete viewpoints and a set of environment maps is created. Since not all samplings can be covered by the limited number of training images, synthesized views are created from other viewpoints using computer graphics rendering techniques.[1] Next, affine covariant features [27] are extracted from the environment maps. In our experiments we use a variant of Hessian- and Harris-affine detector introduced in [15]. We also tested the scale and rotation invariant SIFT detector [10] (see section 4). We then select "good" feature regions which are characterized by their detection repeatability and accuracy. The basic measure of accuracy and repeatability is based on the relative amount of overlap between the detected regions in the environment maps and the respective reference regions projected onto that environment map using the ground truth transformation. The reference regions can be determined e.g. from the parallel views to the corresponding feature region on the object model (*model region*). This *overlap error* is defined as the error in the image area covered by the respective regions [15].

For each model region a *view set* is the set of its appearances in the environment maps from all possible viewpoints (see Fig. 2(b)). Depending on the 3D structure of the target object a model region may be clearly visible only from certain viewpoints in the scene. We create for each model feature a *similarity map* by comparing it with the corresponding extracted features. As a similarity measure we use the Mahalanobis distance between the respective SIFT descrip-

[1] Due to complexity of the target object and the sampling rate this can be a time consuming procedure. However, this does not affect the computational cost of the system at run-time since this can be done off-line.

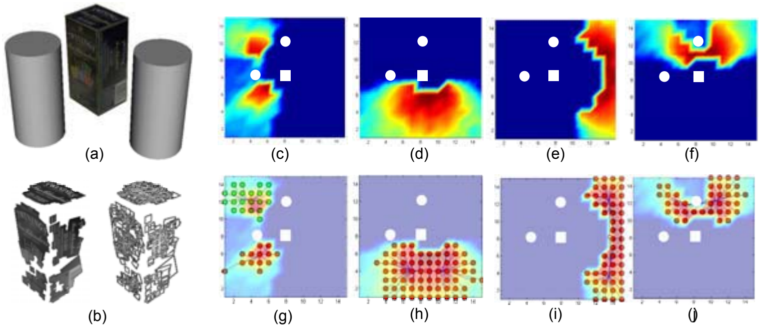


Fig. 3. Experiments with simulated data. (a) The virtual model of the object. (b) The extracted features on the model. (c)-(f) Top-down view of a subset of the similarity maps. (g)-(j) The clustered view sets using mean-shift algorithm.

tors. For each model region the respective similarity map represents its visibility distribution. This analysis can also be used to remove the repetitive features visible from the same viewpoints in order to keep the more distinctive features for matching. Based on the similarity maps of each model region we cluster groups of viewpoints together using the mean-shift algorithm [28]. The clustered viewpoints for a model region m_j are $W(m_j) = \{v_{j,k} \in \Re^3 | 0 < k \leq N_j\}$, where $v_{j,k}$ is a viewpoint of that region. Figure 3 shows some results of a simulated scene including a box and two cylinders. The faces of the box are rendered with the texture obtained from a real tea box. Figure 3(c)-(f) show top down views of a subset of the similarity maps of four patches selected from each side of the box. Note how the presence of an occluding object (cylinders) is reflected in the similarity maps. The respective view sets determined by mean shift clustering are shown in Fig. 3(g)-(j).

3.2 Learning the Statistical Representation

This section describes a method to incorporate multiple view descriptors of each view set into our statistical model. We use the PCA-SIFT descriptor [29] for a more compact representation (e.g. first 32 components). To minimize the impact of variations of illumination, especially between the real and synthesized images, the descriptor vectors are normalized to unit magnitude. The image gradient vectors $g_{i,j}$ are projected into the feature space to a feature vector $e_{i,j}$.

We suppose that the distribution of the gradient vectors is Gaussian for the carefully selected features as described in the previous section. For each region we take k samples from the respective environment maps so that the distribution of their feature vectors $e_{i,j}$ for $0 < j \leq K$ in the feature space is Gaussian. To ensure the Gaussian distribution of the gradient vectors for each view set we apply the χ^2 test for a maximal number of samples. If the χ^2 test fails after a certain number of samplings for a region, the region will be considered as not reliable enough and will be excluded. For each input view set V_i we then learn the covariance matrix Σ_i and the mean μ_i of the distribution.

3.3 Matching as a Classification Problem

Matching is the task to find groups of corresponding pairs between the regions extracted from the model and test image, that are consistent with both appearance and geometric constraints. The matching problem can be formulated as a classification problem [19]. Our goal is to construct a classifier so that the misclassification rate is low. From the test image, the features are extracted in the same manner as in the learning stage and their gradient image vectors are computed. The descriptors are then projected into feature space using PCA (bold dots in Fig. 1). We use the Bayesian classifier to decide whether a test descriptor belongs to a view set class or not. Let $C = \{C_1, ..., C_N\}$ be the set of all classes representing the view sets and let F denote the set of 2D-features $F = \{f_1, ..., f_K\}$ extracted from the test image. Using the Bayesian rule the *a posteriori probability* $P(C_i|f_j)$ for a test feature f_j that it belongs to the class C_i is calculated as

$$P(C_i|f_j) = \frac{p(f_j|C_i)P(C_i)}{\sum_{k=1}^{N} p(f_j|C_k)P(C_k)}. \tag{1}$$

We compute for each test descriptor the a posteriori probability of all classes and select candidate matches using thresholding. Let $m(f_j)$ be the respective set of most probable potential matches $m(f_j) = \{C_i|P(C_i|f_j) \geq T\}$. The purpose of this threshold is only to accelerate the run-time matching and not to consider matching candidates with low probability. However this threshold is not crucial for the results of pose estimation.

3.4 Pose Estimation Using Geometric Inference

This section describes a method using geometric consistency to constrain the search space for finding candidate matches. For the pose estimation a set of $N \geq 3$ matches are required. In an iterative manner we choose the first match $f_1' \leftrightarrow C_1'$ as the pair of correspondences with the highest confidence:

$$argmax_{\substack{f_k \in F \\ C_l \in C}} P(C_l|f_k).$$

We define V_{C_l} as the set of all classes of regions which should also be visible from the viewpoints where C_l is visible

$$V_{C_l} = \{C_k \in C||W_k \cap W_l| \neq 0\},$$

where W_j is the set of 3D-coordinates of the clustered viewpoints $\{v_{j,k}|0 < k \leq N_j\}$ for which the respective model region is visible (see building environment maps, Section 3.1).

Assuming the first candidate match is correct, the second match $f_2' \leftrightarrow C_2'$ is chosen only from the respective set of visible regions. Therefore after each match selection the search area is constrained to visibility of those regions based on previous patches. In general the k^{th} candidate match $f'_k \leftrightarrow C'_k, 1 < k \leq N$ is selected in a deterministic manner

$$(f'_k, C'_k) = argmax_{\substack{f_k \in F \backslash \{f_1, ..., f_{k-1}\} \\ C_k \in \bigcap_{l=1}^{k-1} V_{C'_l}}} P(C_k | f_k).$$

The termination criteria is defined based on the back-projected overlap error (see Section 3.1) in the test image. This algorithm can be implemented in different ways. One way is a recursive implementation with an interpretation tree where the nodes are visited in the depth-first manner. The depth is the number of required matches N for the pose estimation method. This algorithm has a lower complexity as the results will show, than the plain version of RANSAC or the "exhaustive" version where all pairs of candidate matches are examined.

4 Experimental Results

The proposed method has been tested in a series of experiments using virtual and real objects. Due to the space limitations we only present a subset of the results using real objects. The off-line learning process uses ImageModeler from RealViz [30] to obtain a 3D model.[2] Our experimental setup consists of a target object and a commonly available FireWire camera (Fire-I). The camera is internally calibrated and lens distortions are corrected using the Tsai's algorithm [31].

We conducted a set of experiments to analyze the functional and the timing performance of our approach. The results were compared against a conventional approach based solely on 2D key frames. Our approach requires an input consisting of a set of images (or key frames) of the target object. One target object is shown in Fig. 4(a). The key frames were calibrated. We used a calibration object (a known set of markers) for automatically calibrating the views. These markers were used to compute the ground truth for evaluating the matching results on test frames as well.

In the first experiment, we analyzed the functional performance against view point variations for the same scene but under uncontrolled lighting. The images were taken by a moving camera around the object. For the sake of clarity of presentation, we show a subset of 19 test images from this sequence with additional two images as key frames (see Fig. 4(a)-(b),(d)). All those images were calibrated as explained above. Fig. 4(d) shows some metrics we used to compare these results. One measure of performance is the final size of the representation (number of features in the database) used for both methods indicated by the two straight lines. With increasing number of key frames the size of the database in the conventional case would increase linearly with the number of key frames. In contrast, our method keeps fewer features in the 2D-3D database after careful implicit analysis of their planarity, visibility and detection repeatability. The database size in our method is proportional to the scene complexity not the

[2] The accuracy requirements depend on the underlying pose estimation algorithms, the object size and the imaging device.

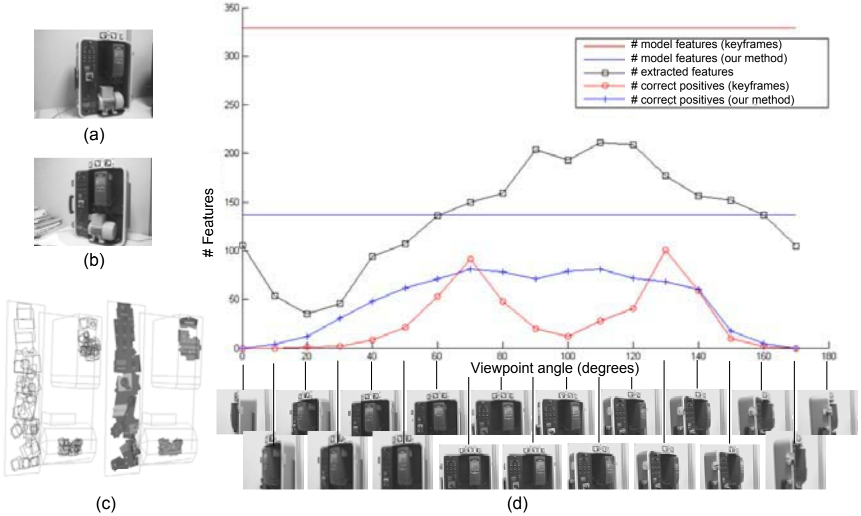


Fig. 4. Experiments with real data. (a)-(b) The calibrated key frames. (c) The set of most visible patches extracted on the model based on the statistical analysis using the similarity maps. (d) Metrics used to compare the results (see text).

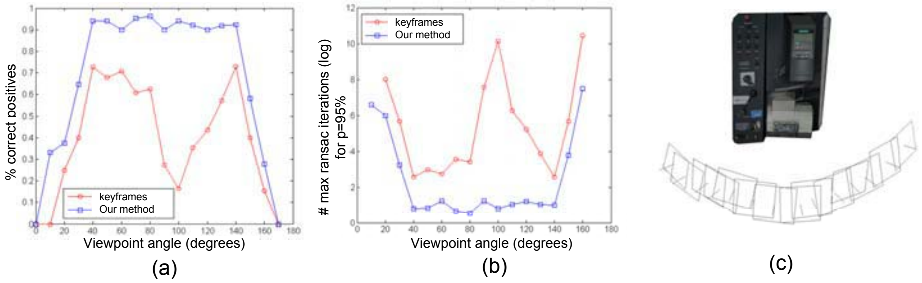


Fig. 5. Experiments with real data. (a)-(b) Performance evaluation (see text). (c) Visualization of the pose estimation results.

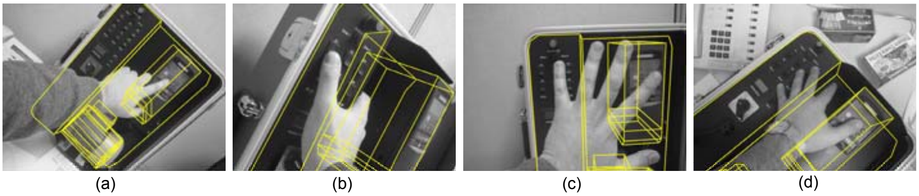


Fig. 6. Experiment 1: Control Box. Pose estimation results on test images.

number of available key frames. This is an important property for the scalability of the system for more complex objects. Fig. 4(d) also shows the number of extracted features and the number of correct matches found by both methods

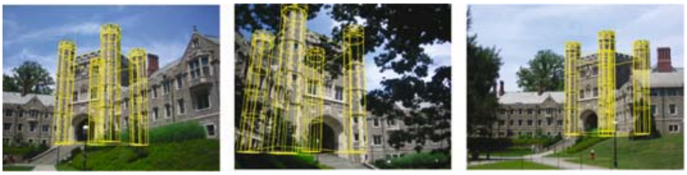


Fig. 7. Experiment 2: Blair Tower. Pose estimation results on test images.

for each of the 19 test images. It should be noted that, near the two key frames our method obtains less correct matches compared to the conventional method. This is due to the fact that our representation generalizes the extracted features whereas the conventional methods keeps them as they are. The generalization has the cost of missing some of the features in the images closer to the key frames. On the other hand, the generalization helps to correctly match more features in disparate test views.

Complexity and performance of robust pose estimation methods like RANSAC are dependent not on the number of correct matches but the ratio between correct and false matches. Fig. 5(a) shows the percentage of correct matches vs the viewing angle for the proposed method and the conventional approach. Although near the key frames our method obtains fewer matches, it has a higher percentage of correct positives. As a result of this and the visibility constraints used our method needs only a few RANSAC iterations for pose estimation. This brings us to the timing performance of the matching methods. We use a more complex matching method than the conventional one. Therefore, each individual match costs more. However, with increasing complexity of the target object with respect to self-occlusions our representation becomes more efficient. Fig. 5(b) shows the respective maximal number of iterations needed (logarithmic scale) for RANSAC based pose estimation with a confidence probability of 95%. Fig. 5(c) shows a visualization of the pose estimation results. We obtain up to five folds speed-up compared to the exhaustive RANSAC method. Our non-optimized implementation needs about 0.3 to 0.6 second compared to 2.5 seconds for the conventional approach. In Fig. 6 (a)-(d) more results are shown for experiments using test images with occlusions, cluttered background

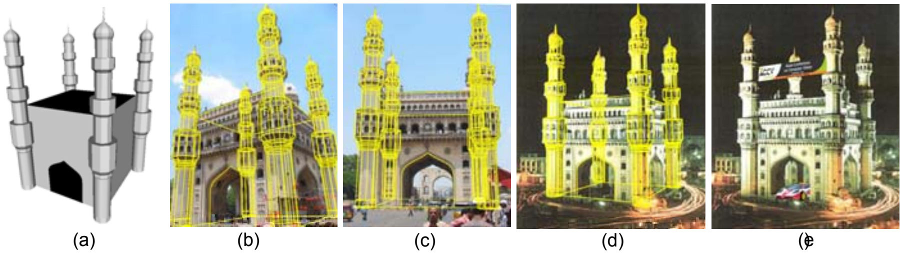


Fig. 8. Experiment 3: Char Minar. (a) 3D model. (b)-(d) Pose estimation results on test images, and with virtual objects (e).

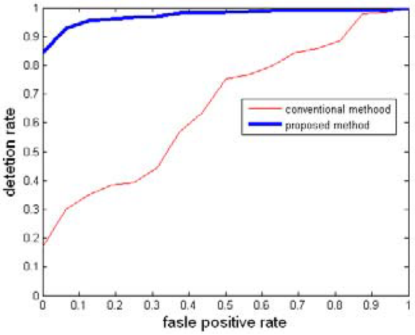


Fig. 9. Performance evaluation: ROC plot

and illumination changes. The detection results are quite robust and the estimated pose is accurate enough to initialize our real-time 3D tracker [6]. Fig. 8 and 7 show the results of two other experiments in outdoor environments. We used each time two images to build a coarse 3D model and applied our method to several test images.

The performance of the matching part of our system was evaluated by processing all pairs of object model and test images, and counting the number of established matches. Fig. 9 shows the ROC curve that depicts the detection rate vs false-positive rate, while varying the detection threshold T. Compared to the keyframe-based approach the proposed approach performs very well and achieves 97% detection with 5% false-positives.

5 Conclusions

This paper addressed the problem of real-time object detection for pose estimation. The major contribution of this paper is the integration of the known 3D geometry of the target model during both matching and pose estimation steps. This is achieved by a statistical analysis of the appearances distribution of model patches in the viewing space. Instead of the local planarity assumption used in previous approaches, our proposed method is able to learn the visibility distribution of the variations in the local descriptors considering their known geometry.

Acknowledgments. We would like to thank D. Lowe for providing his implementation of the SIFT Keypoint Detector. We also would like to thank G. Klinker, Y. Tsin and V. Ramesh for helpful discussions during the course of this work.

References

1. Dementhon, D., Davis, L.S.: Model-based object pose in 25 lines of code. ECCV (1992)
2. Pollefeys, M., Koch, R., Van Gool, L.: Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. ICCV (1998)

3. Nister, D.: An efficient solution to the five-point relative pose problem. CVPR (2003)
4. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)
5. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3d tracking using online and offline information. PAMI (2004)
6. Genc, Y., Riedel, S., Souvannavong, F., Akinlar, C., Navab, N.: Marker-less tracking for ar: A learning-based approach. ISMAR (2002)
7. Davison, A., Murray, D.: Simultaneous localization and map-building using active vision for a robot. PAMI (2002)
8. Ferrari, V., Tuytelaars, T., Van Gool, L.: Integrating multiple model views for object recognition. CVPR (2004)
9. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: Segmenting, modeling, and matching video clips containing multiple moving objects. CVPR (2004)
10. Lowe, D.: Distinctive image features from scale-invariant key points. IJCV (2004)
11. Meltzer, J., Soatto, S., Yang, M.H., Gupta, R.: Multiple view feature descriptors from image sequences via kernel principal component analysis. ECCV (2004)
12. Schmid, C., Mohr, R.: Local gray value invariants for image retrieval. PAMI (1997)
13. Lowe, D.G.: Object recognition from local scale-invariant features. ICCV (1999)
14. Van Gool, L., Moons, T., Ungureanu, D.: Affine/photometric invariants for planar intensity patters. ECCV (1996)
15. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. ECCV (2002)
16. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or how do i organize my holiday snaps?. ECCV (2002)
17. Nayar, S.K., Nene, S.A., Murase, H.: Real-time 100 object recognition system. PAMI (1996)
18. Li, Y., Tsin, Y., Genc, Y., Kanade, T.: Object detection using 2d spatial ordering constraints. CVPR (2005)
19. Lepetit, V., Pilet, J., Fua, P.: Point matching as a classification problem for fast and robust object pose estimation. CVPR (2004)
20. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using affine-invariant patches and multi view spatial constraints. CVPR (2003)
21. Tuytelaars, T., VanGool, L.: Wide baseline stereo matching based on local, affinely invariant regions. BMVC (2000)
22. Allezard, N., Dhome, M., Jurie, F.: Recognition of 3d textured objects by mixing view-based and model-based representations. ICPR (2000)
23. Jurie, F.: Solution of the simultaneous pose and correspondence problem using gaussian error model. CVIU (1999)
24. Mindru, F., Moons, T., VanGool, L.: Recognizing color patterns irrespective of viewpoint and illumination. CVPR (1999)
25. Lepetit, V., Lager, P., Fua, P.: Randomized trees for real-time keypoint recognition. CVPR (2005)
26. Adelson, E.H., Bergen, J.R.: Computational models of visual processing, chapter 1: The plenoptic function and the elements of early vision. The MIT Press, Cambridge (1991)
27. Mikolajczyk, K., Tuytelaars, T., Schmid, C. Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. IJCV (2004)

28. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI (2002)
29. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. CVPR (2004)
30. RealViz. (www.realviz.com)
31. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using of the shelf tv cameras. IEEE Journal of Robotics and Automation (1987)

Efficient 3D Face Reconstruction from a Single 2D Image by Combining Statistical and Geometrical Information

Shu-Fan Wang and Shang-Hong Lai

Department of Computer Science, National Tsing Hua University,
101, Kuang Fu Rd, Sec.2, HsinChu, Taiwan 300, ROC
lai@cs.nthu.edu.tw
http://cv.cs.nthu.edu.tw/index.htm

Abstract. In this paper, we present an efficient algorithm for reconstructing 3D head model from a single 2D image based on using a 3D eigenhead model. This system is composed of two components, offline training of the eigenhead model and online reconstruction of a 3D head model. For the first part, we propose a new 3D head alignment algorithm based on an iterative coarse-to-fine scheme to establish dense point correspondences between 3D head model in the cylindrical coordinate to align the 3D head models in the training data set. In addition, we apply the radial basis function technique to establish dense correspondences between each 3D face model and a reference face model, followed by the principal component analysis technique to compute the statistical eigenhead model. For the 3D face reconstruction from a single image, the proposed algorithm finds the best linear combination of the eigenhead bases that minimizes an energy function composed of distances between the corresponding facial feature points and a one-way partial Haussdorf distance between the facial contours in the image domain. This energy minimization is accomplished by the iterative Levenberg-Marquardt algorithm with the initial guess determined by solving a linear system derived from the image projection constraints for the corresponding facial feature points. Experimental results show that the proposed 3D face reconstruction algorithm provides satisfactory results and takes less than 10 seconds on a regular PC.

1 Introduction

Model-based statistical techniques have been widely used in various fields in computer vision. For example, Turk and Pentland [1] proposed the eigen-face technique to recognize faces from images by projecting face images onto a PCA subspace. This PCA technique extracts features that are critical for face recognition. Later, Cootes and Taylor [2] presented a statistical shape modeling technique to describe 2D shape of an object by an Active Shape Model (ASM). In addition, Blanz and Vetter [3] developed a morphable model for modeling textured 3D faces, which is accomplished by transforming the shape and texture into a vector space representation. They used this morphable model to reconstruct 3D face model from a single image. More recently, this 3D morphable model was also applied to achieve high-accuracy face recognition with pose variations and a wide range of illumination changes.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 427 – 436, 2006.

In most statistical techniques, one of the most crucial aspects is to represent object shapes by a set of landmark points. The statistical principal component analysis of a training dataset of landmark points leads to a Point Distribution Model (PDM) [4]. To correctly construct the PDM, the landmark points need to be properly aligned and the correspondences of landmark points need to be established properly. Many approaches have been proposed to register the landmark points. Some methods [3, 5, 6] resolved the 3D point correspondence problem by applying 2D registration techniques on a 2D projected space. Since the human head shape is close to a 3D ellipsoid, the 3D face data sets can be easily mapped to a cylindrical coordinate. Therefore, the correspondences between 3D point data sets can be established by minimizing the distances between point data sets or by optical flow computation on texture intensity. In this paper, we proposed a more accurate method to find correspondences between 3D head models by transforming the data sets onto a cylindrical coordinate. The proposed method is primarily based on modifying the method proposed in [5] to achieve higher accuracy in finding correspondences of 3D face data points.

Most of previous 3D face reconstruction techniques require more than one image to achieve satisfactory 3D human face modeling. Although there are several different approaches for 3D reconstruction from a single image, such as shape from focus and shape from texture, these approaches are not well suited for 3D face reconstruction due to very limited texture information in the human face images. Another approach for 3D face reconstruction from a single image is to simplify the problem by using a prior statistical head model. For example, Atick et al. [11] combined the shape from shading constraint with the prior eigenhead model to reconstruct a 3D face model by minimizing the corresponding energy function. Recently, Blanz and Vetter [3] proposed an algorithm for 3D face model reconstruction by minimizing an energy function of discrepancies between the face image and the corresponding image rendered from a morphable 3D head model under a suitable illumination condition. However, these two methods are quite computationally expensive since not only the 3D head model but also the illumination conditions are involved in the minimization. In addition, they require a good initial guess to converge to a satisfactory solution.

In this paper, we propose a novel 3D head model reconstruction algorithm based on a prior 3D eigenhead model learned from a set of 3D face models. In contrast to the aforementioned previous methods based on a prior statistical 3D head model, the proposed algorithm employs the geometric constraints derived from the distance between the corresponding facial feature points in the image projection space as well as the Hausdorff distance between the facial silhouettes computed from the face image and the projection of the estimated 3D head model. Our algorithm based on geometric constraints is much more computationally efficient since the energy function to be minimized is independent of the illumination condition.

2 PDM Construction

The PDM provides the prior knowledge of 3D human face models. The training 3D face models are collected from two sources. The first part contains 69 face models acquired from a 3D laser scanner. There are 65 males and 4 females with ages between 22 and 25. In the second part of the 3D data set, we used 55 face models, which contain 43 male and 12 female of ages between 18 and 40, provided by GAVAB [11]. The flow chart is shown in Figure 1.

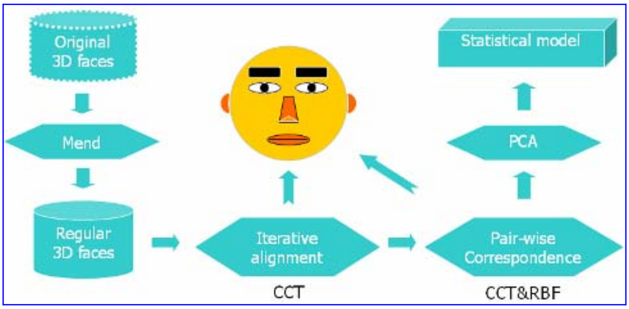


Fig. 1. Flow diagram of the PDM construction. (CCT: Cylindrical Coordinate Transformation).

2.1 Iterative Alignment

In order to estimate correspondences between 3D face models, we need to first align the faces to a reference face model. The iterative closest point (ICP) method has been used to determine the pose parameters for aligning 3D models. However, this approach normally requires a good initial solution to converge to the global optimal solution. The other approach [5] aligns 3D head models from a small set of corresponding 3D facial feature points which are selected manually. For this approach, the manual selection errors will be propagated into the estimation of alignment parameters, and it does not account for the alignment of the whole 3D face data points.

To obtain dense 3D face data correspondences, we use a complete head model to be the reference model as shown in Figure 2(a). Then, we manually select 16 landmark points on each 3D face model as depicted in Figure 2(b). From the correspondences of these 16 landmark points, we can determine an initial estimate of the scale, rotation and translation parameters by minimizing the following energy function,

$$\arg\min_{R,t} \sum_{i=1}^{n} \left\| x_i^r - (Rx_i + t) \right\|^2 \tag{1}$$

where x_i means i-th feature point in a face data set, x_i^r denotes the feature points in the reference 3D face model, and n is the number of feature points. Thus, the initial 3D alignment is obtained by minimizes the above energy function [8].

After the initial alignment of 3D face models, we find more point correspondences automatically to refine the 3D alignment. Instead of choosing the closest points between 3D head models for point correspondence, we do it by transforming the 3D head models into a cylindrical coordinate by equation (2) as depicted in Figure 3.

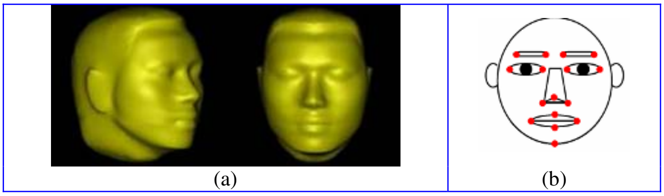


Fig. 2. The generic 3D head model and the corresponding feature points. (a) Generic head model in two views and (b) the 16 facial landmark points used in the system.

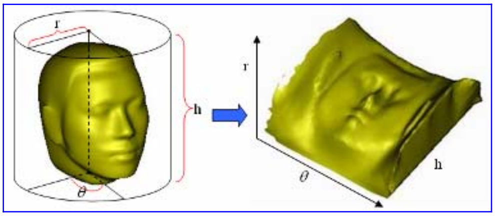


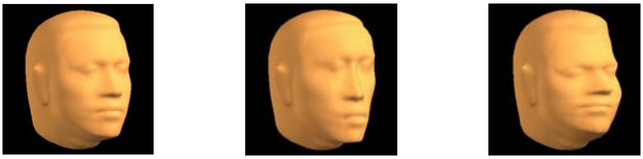
Fig. 3. Cylindrical coordinate transformation (CCT): the 3D face model is transformed to a 2D cylindrical coordinate space

$$(\theta, h, r) = \begin{pmatrix} \tan^{-1}\left(\frac{X-X_C}{\sqrt{(X-X_C)^2+(Z-Z_C)^2}}, \frac{Z-Z_C}{\sqrt{(X-X_C)^2+(Z-Z_C)^2}}\right), Y - Y_C, \\ \sqrt{(X-X_C)^2+(Z-Z_C)^2} \end{pmatrix} \quad (2)$$

Then the corresponding pairs can be determined easily. We first discretize the cylindrical coordinate space (θ, h) with a regular grid. Then we find the correspondence for each point of the reference face model from interpolation of the sample head model at the same (θ, h) coordinate in the cylindrical space. Thus, dense correspondence between the 3D face models can be established to refine the 3D model alignment. The proposed alignment algorithm iteratively refines the 3D alignment.

2.2 Refined Correspondence Via RBF Transformation

After the above iterative 3D alignment, each face model is aligned to a reference face model but we do not simply use the correspondence pairs as the statistical training data sets, because 3D rigid transformation used in the above 3D face model alignment cannot provide satisfactory matching of facial feature points. Therefore, we apply the Radial Basis Function approximation [7] to determine an elastic transformation that interpolates the pairs of correspondence landmark points. We warp each 3D face model to the reference face model by RBF based on the landmark points. Therefore, the feature points between models will be matched exactly and the transformed locations of all the vertices are computed in the Euclidean space, then they are mapped to the cylindrical space to find the corresponding points in the other face model. Then we can apply the PCA to find the major eigen-modes of variations in the 125 training data set. Some examples are depicted in Figure 4.



Mean model $\overline{M}$ $\qquad$ $\overline{M} - 8\sum_{i=1}^{5}\sqrt{\lambda_i}V_i$ $\qquad$ $\overline{M} + 8\sum_{i=1}^{5}\sqrt{\lambda_i}V_i$

Fig. 4. Examples of the linear combinations of 3D head eigen-modes

3 3D Face Reconstructions

In this section, we present an algorithm for reconstructing the 3D head model from a single face image by using the eigenhead model, which is learned from a set of scanned 3D head data. The proposed algorithm consists of two stages; namely, initial 3D face estimation from 2D facial feature correspondences and refined face model reconstruction combining facial feature and contour matching, both using the 3D eigenhead model. The flow chart is shown in Figure 5.

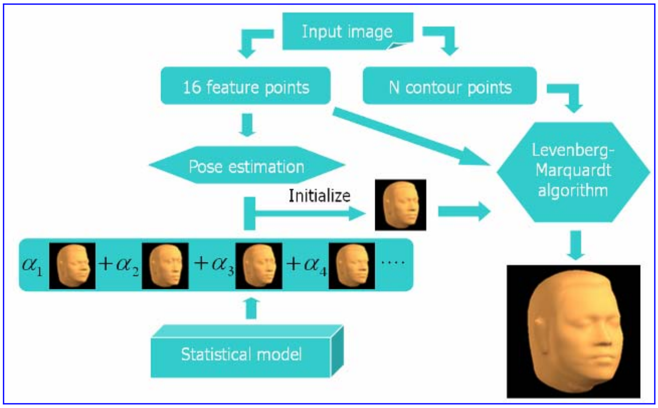


Fig. 5. Flow diagram of 3D face reconstruction. We apply LM algorithm to optimize the final results by considering both feature points and contour information simultaneously.

3.1 Initialize 3D Face from 2D Feature Information

In this work, we focused on the 3D reconstruction from a near-frontal human face image. We assume that the 16 facial feature points have been extracted. These feature points include four corners of eyebrows, four corners of eyes, tip and two sides of nose, corners of mouth, top of upper lip, bottom of lower lip shown as Figure 2(b).

To initialize a 3D face model, we first estimate the pose of human head by 16 feature points, and the pose information can be obtained by minimizing the energy function:

$$E = \sum_{i=1}^{n} \left\| u_i - (sR\bar{x}_i + t) \right\| \tag{3}$$

where u_i is the i-th feature point on a 2D image, and x_i denotes the i-th feature point on the mean face. The scale s, rotation matrix R and translation vector t can be estimated by minimizing the total re-projection errors with LM algorithm. We restrict the rotation angle within 5 degrees since we are mainly interested in the frontal face image.

Let a 3D face model M be represented by a mean head model and a linear combination of eigenhead basis vectors (v_1, v_2, v_3, …, v_n) as follows:

$$M = \overline{M} + \sum_{i=1}^{e} \alpha_i v_i \tag{4}$$

where $\alpha_1,\alpha_2,\ldots,\alpha_n$ are the weights associated with the eigenhead basis vectors.

To obtain an initial face model from the 2D-3D correspondences of the 16 facial feature points, we can formulate this model estimation problem as the linear system:

$$\begin{bmatrix} \bar{x}_1 & v_{1,1x} & v_{1,2x} & .... & v_{1,ex} & 1\,0 \\ \bar{y}_1 & v_{1,1y} & v_{1,2x} & .... & v_{1.ey} & 0\,1 \\ : & : & : & .... & : & : \\ \bar{y}_{fn} & v_{fn,1x} & v_{fn,2x} & .... & v_{fn,ey} & 0\,1 \end{bmatrix}_{2fn\times(3+e)} \begin{bmatrix} s \\ \alpha_1 \\ \vdots \\ \alpha_e \\ t_x \\ t_y \end{bmatrix}_{(3+e)\times 1} = R^{-1} \begin{bmatrix} u_{1,x} \\ u_{1,y} \\ u_{2,x} \\ u_{2,y} \\ \vdots \\ u_{fn,x} \\ u_{fn,y} \end{bmatrix}_{(2fn)\times 1} \tag{5}$$

where R is the concatenation of the estimated rotation matrix, s is the scale, $v_{i,j}$ denotes the i-th element of j-th eigenhead basis vector, n is the total number of eigenmodes used in the face reconstruction, and u_i is the i-th 2D feature point coordinate. We can simply solve the above linear system to obtain the weights of the eigen-head basis vectors, thus an initial 3D head model is determined.

3.2 Detail Refinement of 3D Face Model

In this refinement step, we apply the initial 3D face model and combine the feature points and contour information to reconstruct a 3D human face model by using LM optimization. One of the most reliable information for 3D face reconstruction from a single image is the 2D face contour. Based on the 16 feature points, we simply apply a spline-based contour extraction method to detect the face contour in a face image by fitting a contour with local maximal image gradients. In addition, we also proposed a 3D face contour extraction algorithm given as follows:

1. Estimate the normal direction of each vertex by computing the smoothed normal direction in the 3D face surface.
2. The vertices with normal direction orthogonal to the viewing direction are selected as the contour candidates.
3. Split the space of all contour candidates into several bins, and choose one from each bin with the maximum z value as the 3D contour control point.
4. Compute a spline from these control points as the final 3D contour and project the dense points on the spline curve to 2D image.

After the extraction of the projected contour of 3D face model, the problem of how to define the distance between the re-projected contour and the extracted 2D image contour is critical. Here we apply a modified Hausdorff distance to measure the distance between two face contours, which is called partially one-way Hausdorff distance:

$$H(A,B) = \underset{a\in A}{K^{th}} \min_{b\in B} \|a-b\| \tag{6}$$

where the K-th largest value of the minimum distance from the location of point set A to point set B is used as the distance between the two point sets. We use a fractional value f to determine an appropriate value K to be $f = \underset{a\in A}{K^{th}} \Big/ |A|$. Note that $f = 0.9$ to prevent some outlier effect. The idea of is depicted in Figure 6.

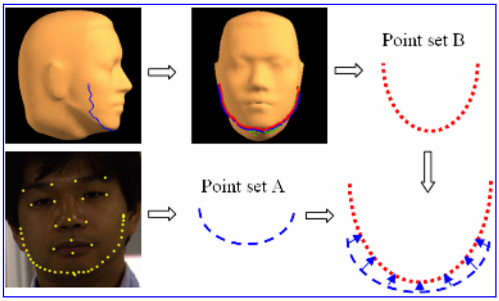


Fig. 6. Partially One-Way Hausdorff Distance: Sparse point set A contains sparse points on a 2D contour, and Dense point set B contains 2D re-projected contour points from a 3D model

Because the detected 3D face contour is a set of dense points, we only compute the Hausdorff Distance in the way from 2D image contour to the re-projected contour as depicted in Figure 6. This distance becomes an additional error term:

$$E = \sum_{i=1}^{16} \left\| u_i - (sR\bar{x}_i + t) \right\| + H(C_I, C_{pj}) \tag{7}$$

To be more specific, the 3D face model reconstruction problem is resolved by minimizing the following energy function:

$$E = \sum_{i=1}^{n} \left\| u_i - (sR(\bar{x}_i + \sum_{j=1}^{e} \alpha_j v_{j,i}) + t) \right\| + H(C_I, C_{pj}(M)) \tag{8}$$

where C_I is the contour point set extracted from a 2D face image, and C_{pj} means the re-projected contour point sets computed from the reconstructed 3D face model *M*. We take both the matching of facial feature points and facial contours into account simultaneously, and minimize the total error by using the LM algorithm [9].

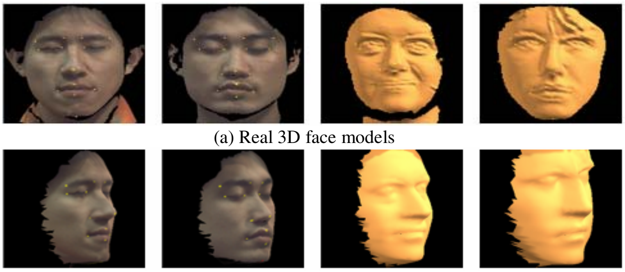
4 Experimental Results

We implemented the proposed system in C++ language and all experiments are conducted on a PC with Intel Pentium Ⅳ 2.8GHz CPU with 504MB RAM. To evaluate the accuracy of the 3D face reconstruction, we define the following error measures.

E_v^{3D}	: Average Euclidean distance of all vertices between original 3D face model and reconstructed 3D face model. The 3D model is fit into a cube with edge 360 in 3D.
E_f^{2D}	: The average error between image and re-projected 2D feature points in frontal view.
E_c^{2D}	: Partially one-way Hausdorff Distance of 2D contour error with $f = 0.9$.

4.1 Simulation Experiments

We took 4 different 3D face models which are not included in the training set to test the accuracy of reconstructed face models. The test 3D sample models are labeled with 16 feature points in 3D space. We re-project them to 2D images at the frontal view as the input testing images, and reconstruct the 3D face models from the simulated 2D input images of size 300-by-300 pixels, and 3D models are fit into a cube with edge 360 in 3D. The reconstructed 3D faces are shown in Figure 7. Table 1 gives the average errors in 2D and 3D space, and the error are actually relatively small enough.

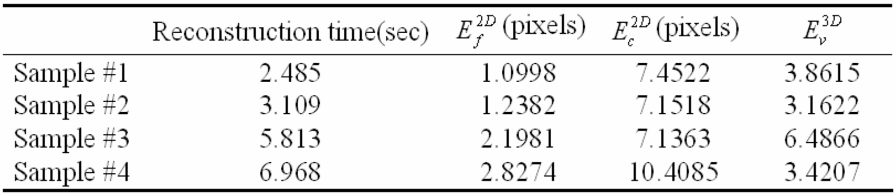


(a) Real 3D face models

(b) Reconstructed 3D face models

Fig. 7. The real testing 3D faces and the reconstructed faces displayed at a near-profile view

Table 1. Reconstruction results on the simulation images



	Reconstruction time(sec)	E_f^{2D} (pixels)	E_c^{2D} (pixels)	E_v^{3D}
Sample #1	2.485	1.0998	7.4522	3.8615
Sample #2	3.109	1.2382	7.1518	3.1622
Sample #3	5.813	2.1981	7.1363	6.4866
Sample #4	6.968	2.8274	10.4085	3.4207

4.2 Real Images Experiments

We tested the proposed 3D face reconstruction algorithm on the face images in the CMU-PIE database [10]. All the individuals in the experiments are not included in the set of training 3D face models for constructing the statistical 3D face model.We selected some frontal face images in the CMU-PIE database to be the testing images

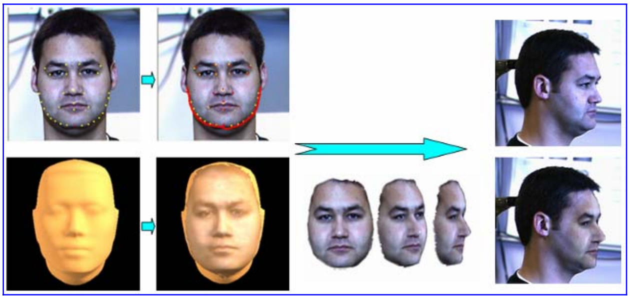


Fig. 8. The reconstruction process: the upper is real and the lower is the reconstructed profile

and the corresponding profile face images are used to perceptually compare the difference between the real face images and the face images rendered from the reconstructed 3D face models as shown in Figure 8 and 9. The re-projection image errors on the reconstructed 3D head models from real images are reported in Table 2. The error ratio to the size of image is relatively small enough and shows the accuracy of our system. The reconstruction time also demonstrate the efficiency of our algorithm.

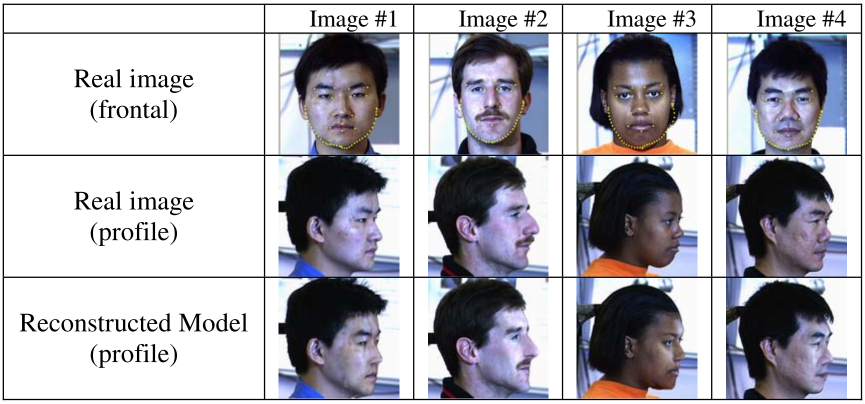


Fig. 9. The comparison of 4 3D face reconstructions

Table 2. Reconstruction results on real images

	Reconstruction time (sec)	E_f^{2D} (pixels) Max	Min	Avg	E_c^{2D} (pixels)
Image #1	8.938	4.5735	0.0339	1.6358	7.6112
Image #2	4.000	4.7782	0.2159	1.8875	11.9887
Image #3	5.905	3.2558	0.2208	1.3528	11.2673
Image #4	6.656	3.7601	0.2654	1.6751	6.3515
Ratio (error/Image size)		0.0108 | 0.0159	0.0001 | 0.0008	0.0042 | 0.0062	0.0211 | 0.0399

5 Conclusions

In this paper, we developed a novel efficient 3D eigenhead-based system to reconstruct a detailed 3D face model from a single 2D face image. An improved 3D face data alignment process was proposed to achieve accurate 3D eigenhead learning. An efficient eigenhead-based 3D reconstruction algorithm was proposed to estimate the 3D face model by combining the statistical and geometrical face information and experimental results on simulated and real images not only show the accuracy of the reconstruction process but also demonstrate its computational efficiency.

References

1. Turk, M., Pentland, A.: Eigenfaces for recognition. Int. J. Cognitive Neuroscience Vol. 3 (1991) 72–86
2. Cootes, B.T., Taylor, C., Cooper, D., Graham, J.: Active Shape Model – their training and application. Computer Vision and Image Understading, 61 (1): (1995) 38–59
3. Blanz, V., Vetter. T.: A Morphable Model for the Synthesis of 3D faces, Computer Graphics Proc. SIGGRAPH, (1999) 187–194
4. Cootes, T.F., Cooper, D.H., Taylor C.J., Graham, J.: A trainable method of parametric shape description, Image and Vision Computing, (1992) 289–294
5. Moghaddam, B., Lee, J.H., Pfister, H., Machiraju, R.: Model-based 3D face Capture with Shape-from-Silhouettes, Int. W., (2003) 20–27
6. Lai, S.H., Chen. Y.L.: Learning a statistical 3D geometric head model, Int. C. SPIE., (2003)
7. Carr J. C., Beatson R.K., Cherrie J. B., Mitchell T. J., Fright W. R., McCallum B. C.: Reconstruction and Representation of 3D Objects with Radial Basis Functions, SIGGRAPH, (2001) 67–76
8. Berthold K. P. Horn, Close-form solution of absolute orientation using unit quaternions, JOSAA, Vol 4, (1987) Issue 4
9. Levenberg. K. A Method for the Solution of Certain Non-linear Problems in Least Squares. Quarterly of Applied Mathematics, 2(2): (1944) 164–168
10. Sim T., Baker S., and Bsat M., The CMU Pose, Illumination, and Expression (PIE) Database, Proc. Int. C. Automatic Face and Gesture Recognition. (2002) 53-58
11. Atick J.J., Griffin P.A., and Redlich A.N., Statistical Approach to Shape from Shading: Reconstruction of 3D Face Surfaces from Single 2D Images, Computation in Neurological Systems, vol. 7, (1996) no. 1
12. http://gavab.escet.urjc.es/

Multiple View Geometry in the Space-Time

Kazutaka Hayakawa and Jun Sato

Department of Computer Science and Engineering, Nagoya Institute of Technology,
Nagoya 466-8555, Japan
hayakawa@hilbert.elcom.nitech.ac.jp,
junsato@nitech.ac.jp

Abstract. In this paper, we analyze multiple view geometry under projections from 4D space to 3D space and show that it can represent multiple view geometry under the projection of space with time. In particular, we show that multifocal tensors defined under space-time projections can be computed from non-rigid object motions viewed from multiple cameras with arbitrary translational motions. We also show that they are very useful for generating images of non-rigid object motions viewed from cameras which have arbitrary translational motions. The method is implemented and tested in real and synthetic images.

1 Introduction

The multiple view geometry is very important for describing the relationship between images taken from multiple cameras and for recovering 3D geometry from images[1, 7]. In the traditional multiple view geometry, the projection from the 3D space to 2D images has been assumed [4]. As a result, the traditional multiple view geometry is limited for describing the case, where enough number of corresponding points are visible from a static configuration of multiple cameras. Recently, some efforts for extending the multiple view geometry for more general point-camera configurations have been made[8, 12]. Wolf et al. [10] studied the multiple view geometry on the projections from N dimensional space to 2D images and showed that it can be used for describing the relationship of multiple views obtained from moving cameras and moving points with constant speed. Unfortunately, the work is limited for the 3D points which move on straight lines with constant speed. Thus the motions of objects are limited.

In this paper we analyze the multiple view geometry under the projection from 4D space to 3D space and show that it can represent multiple view geometry in the case where non-rigid arbitrary motions are viewed from multiple translational cameras. We first analyze affine projections from 4D space to 3D space, and show that we have multilinear relationships for up to 5 views unlike the traditional multilinear relationships. The three view geometry is studied extensively and new trilinear relationship under the projection from 4D space to 3D space is presented. We next show that the newly defined multiple view geometry can be used for describing the relationship between images taken from non-rigid motions viewed from multiple translational cameras. We also show that

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 437–446, 2006.

it is very useful for generating images of non-rigid object motions viewed from arbitrary translational cameras.

2 Affine Projections from 4D Space to 3D Space

We first consider affine projections from 4D space to 3D space. This projection is very important for describing the relationship between the real space-time and the image space-time, and for analyzing the multiple view geometry under space-time projections.

Suppose we have a point in the 4D space, whose homogeneous coordinates are represented by $\mathbf{W} = [W^1, W^2, W^3, W^4, W^5]^\top$. Let $\mathbf{W}$ be projected to a point in the 3D space, whose homogeneous coordinates are represented by $\mathbf{w} = [w^1, w^2, w^3, w^4]^\top$. Then, the extended affine projection from $\mathbf{W}$ to $\mathbf{w}$ can be described as follows:

$$\mathbf{w} \sim \mathbf{P}\mathbf{W} \tag{1}$$

where ($\sim$) denotes equality up to a scale, and $\mathbf{P}$ denotes the following 4×5 matrix:

$$\mathbf{P} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

From (1), we find that the extended affine camera, $\mathbf{P}$, has 15 DOF. In the next section, we consider the multiple view geometry of the extended affine cameras.

3 Affine Multiple View Geometry Under Projections from 4D Space to 3D Space

If we have N extended affine cameras, the geometric DOF of these N cameras is $15N - 20$, since each extended affine camera has 15 DOF and these N cameras are in a single 4D affine space whose DOF is 20. This means two view geometry has 10 DOF, three view geometry has 25 DOF, four view geometry has 40 DOF, and five view geometry has 55 DOF under the extended affine cameras. Unlike the standard multiple view geometry in the 3D space, we have multilinear relationships up to five views for the 4D space, as we will see later.

From (1), we have the following equation for N cameras:

$$\begin{bmatrix} \mathbf{P} & \mathbf{w} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}' & \mathbf{0} & \mathbf{w}' & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}'' & \mathbf{0} & \mathbf{0} & \mathbf{w}'' & \cdots & \mathbf{0} \\ \vdots & & & & & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{W} \\ \lambda \\ \lambda' \\ \lambda'' \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{bmatrix} \tag{3}$$

The left most matrix, $\mathbf{M}$, in (3) is $(4N) \times (5 + N)$, and the determinants of a $(5+N)\times(5+N)$ sub square matrices, $\mathbf{Q}$, of $\mathbf{M}$ constitute multilinear relationships under the extended affine projection as follows:

$$\det \mathbf{Q} = 0 \tag{4}$$

We can choose any $5+N$ rows from $\mathbf{M}$ to constitute $\mathbf{Q}$, but we have to take at least 2 rows from each camera for deriving meaningful N view relationships (note, each camera has 4 rows in $\mathbf{M}$). Thus, the following condition must hold for defining multilinear relationships for N view geometry in the 4D space:

$$5 + N \geq 2N \tag{5}$$

Thus, we find that, unlike the traditional multiple view geometry, the multilinear relationship for 5 views is the maximal linear relationship in the 4D space.

We next consider the minimum number of points required for computing the multifocal tensors. Let us consider M points in the 4D space, and let these points be projected to N affine cameras defined in (1). Then, we have $3MN$ measurements from images, while we have to compute $15N-20+4M$ components for fixing all the geometry in the 4D space. Thus, the following condition must hold for computing the multifocal tensors from images:

$$3MN \geq 15N - 20 + 4M \tag{6}$$

From (6), we find that 5 points are enough for computing multifocal tensors in two, three, four and five views.

3.1 Three View Geometry

We next consider the multiple view geometry of three extended affine cameras. For three views, the sub square matrix $\mathbf{Q}$ is 8×8. From $\det \mathbf{Q} = 0$, we have the following trilinear relationship under extended camera projections:

$$w^i w'^j w''^k \epsilon_{jlmn} \epsilon_{krst} \mathcal{T}_i^{lr} = 0_{mnst} \tag{7}$$

where, each index takes 1, 2, 3 and 4, unlike the standard trilinear relationship. ϵ_{ijkl} denotes a tensor, which takes 1 if the permutation from {i,j,k,l} to {1,2,3,4} is even permutation, and takes -1 if it is odd permutation. $\mathcal{T}_i^{lr}$ is the trifocal tensor for the extended cameras and has the following form:

$$\mathcal{T}_i^{lr} = \epsilon_{iabc} \det \begin{bmatrix} \mathbf{a}^a \\ \mathbf{a}^b \\ \mathbf{a}^c \\ \mathbf{b}^l \\ \mathbf{c}^r \end{bmatrix} \tag{8}$$

where, $\mathbf{a}^i$ denotes the ith row of $\mathbf{P}$, $\mathbf{b}^i$ denotes the ith row of $\mathbf{P}'$ and $\mathbf{c}^i$ denotes the ith row of $\mathbf{P}''$ respectively. The trifocal tensor $\mathcal{T}_i^{lr}$ is $4 \times 4 \times 4$ and has 64 components. If the extended cameras are affine as shown in (1), then 22 of them are equal to 0, and thus we have only 41 free parameters in $\mathcal{T}_i^{lr}$ except a scale ambiguity. On the other hand, (7) provides us $4 \times 4 \times 4 \times 4 = 256$ linear equations on $\mathcal{T}_i^{lr}$, but only 9 of them are linearly independent. Thus, 5 corresponding points are enough for computing $\mathcal{T}_i^{lr}$ from images linearly. This agrees with the result of the analysis in section 3.

3.2 Two View, Four View and Five View Geometry

Similarly, the two view, four view and the five view geometry can also be derived for the extended affine cameras. However, they are practically not so significant, and thus we do not study them in this paper.

4 Multiple View Geometry for Multiple Moving Cameras

Let us consider a single point moving in the 3D space. If the multiple cameras are stationary, we can compute the traditional multifocal tensors[4] from the image motion of this point, and they can be used for constraining image points in arbitrary views and for reconstructing 3D points from images. However, if these cameras are moving independently, the traditional multifocal tensors cannot be computed from the image motion of a single point. Nonetheless, we in this section show that if the camera motions are translational as shown in Fig. 1, the multiple view geometry under extended affine projections can be computed from the image motion of a single point, and they can be used for, for example, generating image motions viewed from arbitrary translational cameras.

We first show that the extended affine cameras shown in (1) can be used for describing non-rigid object motions viewed from stationary multiple cameras. We next show that this camera model can also be used for describing non-rigid object motions viewed from multiple cameras with translational motions of constant speed.

The motions of a point, $\widetilde{\mathbf{X}} = [X, Y, Z]^\top$, in the real space can be considered as a set of points, $\widetilde{\mathbf{W}} = [X, Y, Z, T]^\top$, in a 4D space-time where T denotes time and ($\widetilde{\ }$) denotes inhomogeneous coordinates. The motions in the real space are projected to images, and can be observed as a set of points, $\widetilde{\mathbf{w}} = [x, y, t]^\top$, in a 3D space-time on image motions. Since in general sampling period is different

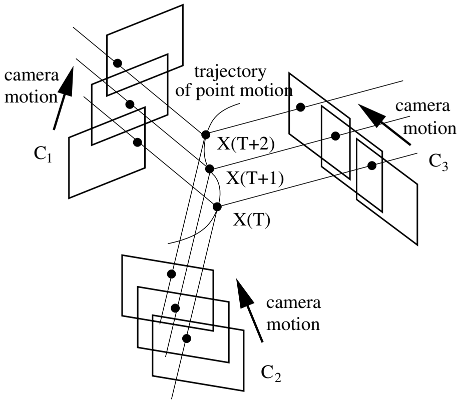


Fig. 1. A moving 3D point and its projections in three translational affine cameras. The multifocal tensor defined under space-time projections can describe the relationship between these image projections.

in each camera, the time t in the image sequence is considered as the affine transformation of original time T. That is $t = aT + b$, where a and b are scalars. Thus, if we assume affine projections in the space axes, the space-time projections can be described by the extended affine cameras shown in (1). For the space-time projections, m_{31}, m_{32} and m_{33} in (1) are always 0, since t is irrelevant to X, Y and Z. If the camera is stationary, m_{14} and m_{24} are also 0. Thus, the projections of non-rigid motions to multiple stationary affine cameras can be described by (1), and thus the multiple view geometry described in section 3 can be applied to this case.

We next show that the multiple view geometry described in section 3 can also be applied for multiple moving cameras. Let us consider a usual affine camera which projects points in 3D space to 2D images. If the translational motions of the affine camera are constant, non-rigid motions are projected to images as follows:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \begin{bmatrix} X(T) - T\Delta X \\ Y(T) - T\Delta Y \\ Z(T) - T\Delta Z \\ 1 \end{bmatrix} \tag{9}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & -a_{11}\Delta X - a_{12}\Delta Y - a_{13}\Delta Z & a_{14} \\ a_{21} & a_{22} & a_{23} & -a_{21}\Delta X - a_{22}\Delta Y - a_{23}\Delta Z & a_{24} \end{bmatrix} \begin{bmatrix} X(T) \\ Y(T) \\ Z(T) \\ T \\ 1 \end{bmatrix} \tag{10}$$

where, $x(t)$ and $y(t)$ denote image coordinates at time t, $X(T)$, $Y(T)$ and $Z(T)$ denote coordinates of a 3D point at time T, and ΔX, ΔY and ΔZ denote camera motions in X, Y and Z axes.

Since the translational motion is constant in each camera, ΔX, ΔY and ΔZ are fixed in each camera. Then, we find, from (10), that the projections of non-rigid motions to multiple cameras with translational motions can also be described by the extended affine cameras shown in (1). Thus the multiple view geometry described in section 3 can also be applied to multiple affine cameras with constant translational motions.

Note, if we have enough number of moving points in the scene, we can compute the traditional multiple view geometry on the multiple moving cameras at each instant. However, if we do not have enough number of moving points in the scene, e.g. if we have just 1 moving point in the scene, we cannot compute the traditional multiple view geometry for multiple moving cameras.

5 Experiments

We next show the results of some experiments. We first show that the trifocal tensor for extended affine cameras can be computed from image motions viewed from arbitrary translational cameras, and can be used for generating the third view from the first and the second views of moving cameras. We next evaluate the stability of extracted trifocal tensors for extended affine cameras. We finally show the results from real images taken from moving cameras.

5.1 Synthetic Image Experiment

In this section, we show by using synthetic images that the trifocal tensors for extended affine cameras can be computed from multiple moving cameras and can be used for recovering the image motions viewed from arbitrary translational cameras.

Fig. 2 shows a 3D configuration of 3 moving cameras and a moving point. The red points show the viewpoints of three cameras, $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3$, before translational motions, and the blue points show their viewpoints after the translational motions. The translational motions of these three cameras are different and are unknown. The black curve shows a locus of a moving point, $\mathbf{X}$. Fig. 3 (a), (b) and (c) show image motions of $\mathbf{X}$ viewed from $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3$ respectively. Note, the original locus of $\mathbf{X}$ is closed in the 3D space as shown in Fig.2, but its loci in images are not closed as shown in Fig.3. This is because the cameras are translating while the 3D point is moving. We added Gaussian image noises with the standard deviation of 1 pixel to all the points on the loci in images.

The green points in Fig.3 (a), (b) and (c) are used for computing an extended trifocal tensor on these three moving cameras. The extended trifocal tensor is used for recovering an image motion in $\mathbf{C}_3$ from image motions in $\mathbf{C}_1$ and $\mathbf{C}_2$. Fig. 3 (d) shows the image motion recovered from Fig. 3 (a) and (b) by using the extracted trifocal tensor. From Fig. 3 (d) and Fig. 3 (c), we find that the recovered image motion is very accurate and stable.

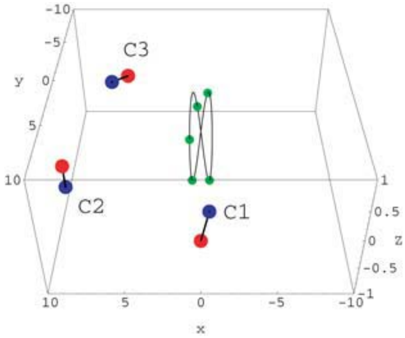
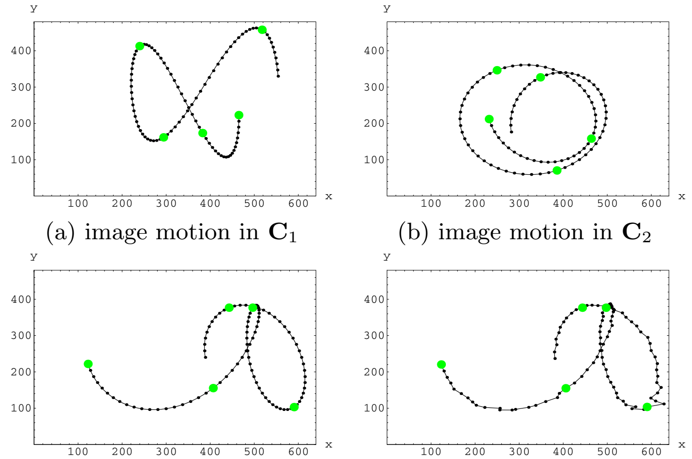


Fig. 2. Three translating cameras and a moving point in the 3D space. The red points show the viewpoints of three cameras before translational motions, and the blue points show those after the translational motions.

5.2 Stability Evaluation

We next show the stability of extracted trifocal tensors under space-time projections. For evaluating the extracted trifocal tensors, we computed reprojection errors derived from the trifocal tensors. In normal cameras, the reprojection errors are computed in 2D images. However, since our cameras are extended to



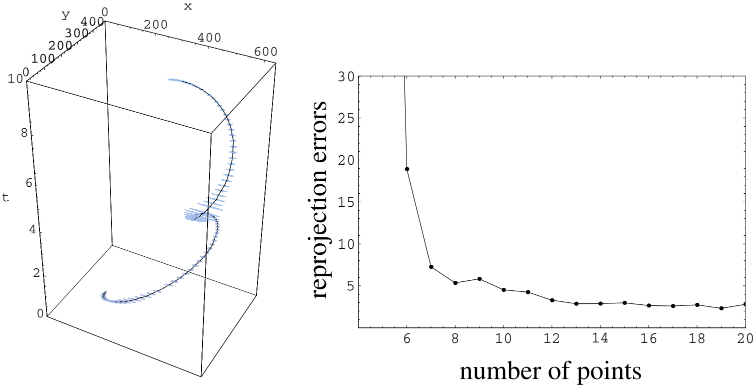
(a) image motion in $\mathbf{C}_1$

(b) image motion in $\mathbf{C}_2$

(c) image motion in $\mathbf{C}_3$

(d) recovered image motion in $\mathbf{C}_3$

Fig. 3. Image motions in three translating cameras, $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3$, and a recovered image motion. The five green points in (a), (b) and (c) are used for computing extended trifocal tensors under space-time projections. (d) shows an image motion in $\mathbf{C}_3$ recovered from the image motions in $\mathbf{C}_1$ and $\mathbf{C}_2$ by using the estimated trifocal tensor.



(a)

(b)

Fig. 4. The stability of recovered loci in the 3D space-time and the reprojection errors. The ellipsoids in (a) show uncertainty bounds of recovered points on the space-time loci. (b) shows the relationship between the number of corresponding points used for computing trifocal tensors and the reprojection errors. The corresponding points are taken randomly from image motions shown in Fig. 3.

the space-time, we compute reprojection errors in the 3D space-time. That is, the reprojection errors are computed from a 3D distance between a true point and a point recovered from the trifocal tensor in the 3D space-time.

We computed trifocal tensors and recovered loci in the 3D space-time in $\mathbf{C}_3$ from image motions in $\mathbf{C}_1$ and $\mathbf{C}_2$ 100 times. Then, the uncertainty bound of each point on the space-time loci is computed. The ellipsoids in Fig. 4 (a) show uncertainty bounds of all the points on the space-time loci in $\mathbf{C}_3$.

We next changed the number of corresponding points for computing trifocal tensors in three views, and evaluated the reprojection errors in the 3D space-time. The reprojection error of a certain number of corresponding points is computed 100 times changing the corresponding points randomly. Fig. 4 (b) shows the relationship between the number of corresponding points and the reprojection errors in the 3D space-time. As we can see, the stability is drastically improved by using a few more points than required. Note, this is the result from the random choice of corresponding points, and thus if we take the corresponding points carefully, the result is much better even from the minimum number of corresponding points.

5.3 Real Image Experiment

We next show the result from a real image experiment. In this experiment, we used two static cameras and one translational camera with a constant speed, and computed trifocal tensors between these three cameras by using a single moving point in the 3D space.

Fig. 5 shows the experimental scene used in this experiment. The camera 1 and camera 3 are stationary, and camera 2 is a moving camera with a constant translation from the left to the right. Since multiple cameras are non-rigid, we can not compute the traditional trifocal tensor of these cameras from a single moving point. Nonetheless we can compute the extended trifocal tensor and can generate image motions in one of three views from the other two views. In this experiment we generated image motions in camera 3 by using image motions in camera 1 and camera 2. Fig. 6 (a) and (b) show image motions of a single point in camera 1 and camera 2 respectively. The trifocal tensor is computed from 5 points on the image motions in three views. These are shown by blue points in

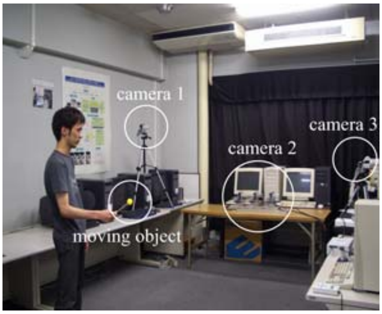
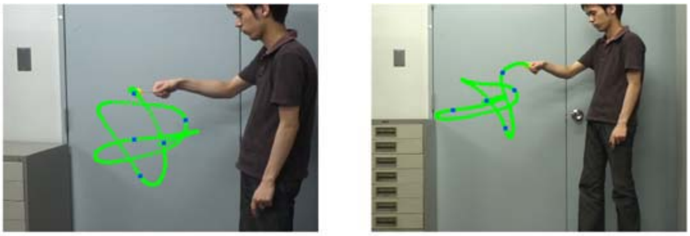
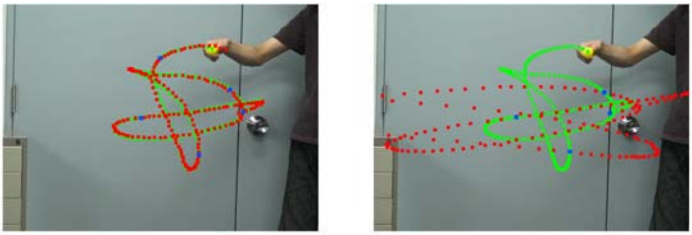


Fig. 5. Real image experiments. The camera 1 and camera 3 are stationary, while the camera 2 translates from the left to the right during the point motions.



(a) image motion in camera 1 (b) image motion in camera 2

Fig. 6. Real image experiments. (a) and (b) show image motions of a single point viewed from camera 1 and camera 2. The 5 blue points in each image show corresponding points used for computing the trifocal tensor. Note, the camera 2 is translating horizontally with a constant speed.



(a) image motions recovered from the extended trifocal tensor (b) image motions recovered from the traditional trifocal tensor

Fig. 7. Image motions in camera 3 recovered from the extended trifocal tensor and the traditional trifocal tensor. The red points in (a) show the image motions recovered from the extended trifocal tensor under space-time projections, and the green points show real image motions observed in camera 3. (b) shows those recovered from the traditional trifocal tensor. The 5 blue points in (a) and 4 blue points in (b) show points used for computing the trifocal tensors.

(a) and (b). The extracted trifocal tensor is used for generating image motions in camera 3 from image motions in camera 1 and 2. The red points in Fig. 7 (a) show image motions in camera 3 generated from the extended trifocal tensor, and the green points show the real image motions viewed from camera 3. As shown in Fig.7 (a), the generated image motions are very accurate even if the camera 2 has unknown translational motions.

To show the advantage of the extended trifocal tensor, we finally show image motions generated from the traditional trifocal tensor, i.e. trifocal tensor defined for projections from 3D space to 2D space. The 4 blue points shown in Fig. 7 (b) are used as corresponding points in three views for computing the traditional affine trifocal tensor. Note, these are the subset of the 5 points used in the previous experiment. The image motion in camera 3 generated from the image motions in camera 1 and 2 by using the extracted traditional trifocal tensor is shown by red points in Fig. 7 (b). As shown in Fig. 7 (b), the generated image motion is very different from the real image motion shown by green points as we expected, and thus we find that the traditional multiple view geometry cannot describe such general situations, while the proposed multiple view geometry can do as shown in Fig. 7 (a).

6 Conclusion

In this paper, we analyzed multiple view geometry under affine projections from 4D space to 3D space, and showed that it can represent multiple view geometry under space-time projections. In particular, we showed that multifocal tensors defined under space-time projections can be computed from non-rigid object motions viewed from multiple cameras with arbitrary translational motions. We also showed that they are very useful for generating images of non-rigid motions viewed from cameras with arbitrary translational motions. The method was implemented and tested by using real image sequences. The stability of extracted trifocal tensors was also evaluated.

References

1. O.D. Faugeras and B. Mourrain, "On the geometry and algebra of the point and line correspondences beterrn N images," in *Proc. 5th International Conference on Computer Vision*, pp. 951-956, 1995.
2. R.I. Hartley, "Multilinear relationship between coordinates of corresponding image points and lines," in *Proc. International Workshop on Computer Vision and Applied Geometry*, 1995.
3. A. Shashua and M. Werman, "Trilinearity of three perspective views and its associated tensor," in *Proc. 5th International Conference on Computer Vision, pp. 920–925*, 1995.
4. R.I. Hartley and A. Zisserman, *Multiple View Geometry*, Cambridge University Press, 2000.
5. A. Heyden, "A common framework for multiple view tensors," in *Proc. 5th European Conference on Computer Vision*, vol. 1, pp. 3–19, 1998.
6. A. Heyden, "Tensorial properties of multiple view constraints," *Mathematical Methods in the Applied Sciences*, vol. 23, pp. 169–202, 2000.
7. O.D. Faugeras and Q.T. Luong, *The Geometry of Multiple Images*, MIT Press, 2001.
8. A. Shashua and L. Wolf, "Homography tensors: On algebraic entities that represent three views of static or moving planar points," in *Proc. 6th European Conference on Computer Vision*, vol. 1, pp. 507–521, 2000.
9. Y. Wexler and A. Shashua, "On the synthesis of dynamic scenes from reference views," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2000.
10. L. Wolf and A. Shashua, "On projection matrices $P^k \rightarrow P^2, k = 3, \cdots, 6$, and their applications in computer vision," in *Proc. 8th International Conference on Computer Vision*, vol. 1, pp. 412–419, 2001.
11. R.I. Hartley and F. Schaffalitzky, "Reconstruction from Projections using Grassman Tensors," in *Proc. 8th European Conference on Computer Vision*, vol. 1, pp. 363–375, 2004.
12. P. Sturm, "Multi-View Geometry for General Camera Models, " in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 206–212, 2005.

Detecting Critical Configuration of Six Points

Yihong Wu and Zhanyi Hu

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, P.O. Box 2728, Beijing 100080, P.R. China
{yhwu, huzy}@nlpr.ia.ac.cn
http://www.nlpr.ia.ac.cn/English/rv

Abstract. When space points and camera optical center lie on a twisted cubic, no matter how many corresponding pairs there are from space points to their image points, camera projection matrix cannot be uniquely determined, in other words, the configuration of camera and space points in this case is critical for camera parameter estimation. In practice, it is important to detect this critical configuration before the estimated camera parameters are used. In this work, a new method is introduced to detect this critical configuration, which is based on an effective criterion function constructed from an invariant relationship between six space points and their corresponding image points. The advantage of this method is that no explicit computation on camera projection matrix or optical center is needed. Simulations show it is quite robust and stable against noise. Experiments on real data show the criterion function can be faithfully trusted for camera parameter estimation.

1 Introduction

Projective geometric invariant plays an important role in computer vision. Since 1994, there have been many studies on the invariant relationship between six space points and their image points [2, 3, 4, 7, 9, 10, 11, 12, 13]. The invariant relationship can be applied to 3D reconstruction, object recognition, robot vision and so on as shown in the literature.

On the other hand, estimation of camera parameters is a key problem in 3D reconstruction. One of the popular methods for this problem is to recover the camera parameters from at least six pairs of image points and their corresponding spatial points with known coordinates [1]. By using this method, many degenerate configurations may occur. There are systematic analyses for these degenerate configurations in Chapter 21 of [6], which consist of two cases: incidence case and non-incidence case. The incidence case is that some of the space points are collinear or coplanar, or some of the space points and the camera optical center are collinear or coplanar. *The non-incidence case is that the space points and the camera optical center lie on a proper twisted cubic*, of which no three space points are collinear and no four space points are coplanar, also no two space points are collinear and no three space points are coplanar with the camera optical center. How to detect these degenerate configurations? This is the problem considered here. For the incidence case, it is easy to detect by determining the

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 447–456, 2006.

linearly dependent relations among the space points or the image points. However, for the non-incidence case, namely the case that the space points and the camera optical center lie on a proper twisted cubic, it is difficult to detect. The method of [16] can detect this degenerate configuration. But, estimation of the camera optical center is needed at first, and also it is sensitive to noise.

Detecting degenerate configurations is important because if a spatial configuration is degenerate mathematically but the noise from the measured image makes it non-degenerate, any estimation under such configuration is useless [15]. For camera parameter estimation, the data from those degenerate configurations just mentioned are critical and can result in dangerous recovered camera parameters.

Our method in this paper can effectively detect the degenerate configuration in the non-incidence case. This degenerate configuration in the non-incidence case (camera and space points lie on a proper twisted cubic) is called *twisted cubic degenerate configuration*, or *twisted cubic configuration* in the following.

By using brackets like in [2,3,4,5], we establish the invariant relationship between six space points and their images under a perspective view for the twisted cubic configuration. This configuration is different from the previous general one [2,3,4,7,9,10,11,12,13]. The established invariant relationship is free of the camera optical center and camera projective matrix. From it, then an algorithm based on a weighed criterion function is proposed to detect the twisted cubic degenerate configuration. Simulations and experiments on real data are performed, which show the proposed algorithm is quite stable against noise and the criterion function is reasonably useful in practice.

The organization of the paper is as follows. Some preliminaries are listed in Section 2. Section 3 reports the invariant relationship between six space points and their images under a perspective view for the twisted cubic configuration, and then elaborates the algorithm to detect the twisted cubic degenerate configuration for camera parameter estimation. Experiments are shown in Section 4, and Section 5 are some conclusions.

2 Preliminaries

In this paper, a bold capital letter denotes either a homogeneous 4-vector or a matrix, a bold small letter denotes a homogeneous 3-vector, a bracket "[ ]" denotes the determinant of vectors in it. And in addition, we assume that no three image points are collinear, no four space points are coplanar (so the brackets on the image and space points are always nonzero).

Under the pinhole camera, a space point $\mathbf{M}_i$ is projected to a point $\mathbf{m}_i$ in the image plane by:

$$s_i \mathbf{m}_i = \mathbf{K}(\mathbf{R}, \mathbf{t})\mathbf{M}_i, \quad i = 1..6, \tag{1}$$

where $\mathbf{K}$ is the 3×3 matrix of camera intrinsic parameters, and $\mathbf{R}, \mathbf{t}$ are a 3×3 rotation matrix and a 3×1 translation vector, s_i is a nonzero scalar. If s_i were zero, then $\mathbf{M}_i$ could not be projected to the image plane. We assume that camera optical center $\mathbf{O}$ and six space points $\mathbf{M}_i$ are not at infinity throughout this paper.

Then under (1), the established relation in [4] between bracket on image points and bracket on space points is:

$$s_i s_j s_k [\mathbf{m}_i, \mathbf{m}_j, \mathbf{m}_k] = \det(\mathbf{K})[\mathbf{M}_\mathrm{i}, \mathbf{M}_\mathrm{j}, \mathbf{M}_\mathrm{k}, \mathbf{O}]. \tag{2}$$

We will use (2) later.

In the following, for the notational convenience, if no ambiguity can be aroused, $\mathbf{M}_i$, $i = 1..6$ will be simply denoted as $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$, and the commas in the brackets will be omitted.

There is a unique proper twisted cubic passing through six space points $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$ with no three collinear and no four coplanar. Any point $\mathbf{X}$ is on this twisted cubic if and only if [16]:

$$\begin{cases} \dfrac{[\mathbf{1246}][\mathbf{1356}]}{[\mathbf{1236}][\mathbf{1456}]} = \dfrac{[\mathbf{124X}][\mathbf{135X}]}{[\mathbf{123X}][\mathbf{145X}]}, \\[2ex] \dfrac{[\mathbf{1246}][\mathbf{2356}]}{[\mathbf{1236}][\mathbf{2456}]} = \dfrac{[\mathbf{124X}][\mathbf{235X}]}{[\mathbf{123X}][\mathbf{245X}]}, \\[2ex] \mathbf{X} \text{ is not on the line through } \mathbf{1},\ \mathbf{2}. \end{cases} \tag{3}$$

The above representation is not unique as a result that the one after a permutation of $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$ is also a representation of the same twisted cubic.

We can see that each bracket in the first equation of (3) has the point $\mathbf{1}$. The geometric meaning of this equation is that $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{X}$ lie on a quadric cone with $\mathbf{1}$ as the vertex [16]. Similarly, the second equation of (3) means that $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{X}$ lie on a quadric cone with $\mathbf{2}$ as the vertex. This is consistent with the theorem in [14] that a twisted cubic can be the intersection of two quadrics.

3 Recognizing Critical Configuration of Six Points

3.1 Invariant Relationship for the Twisted Cubic Configuration

Proposition 1. The camera optical center $\mathbf{O}$ lies on the proper twisted cubic passing through $\mathbf{1}, \mathbf{2}, \mathbf{2}, \mathbf{4}, \mathbf{5}, \mathbf{6}$ if and only if

$$\begin{cases} [\mathbf{m}_1\mathbf{m}_2\mathbf{m}_3][\mathbf{m}_1\mathbf{m}_4\mathbf{m}_5][\mathbf{1246}][\mathbf{1356}] \\ -[\mathbf{m}_1\mathbf{m}_2\mathbf{m}_4][\mathbf{m}_1\mathbf{m}_3\mathbf{m}_5][\mathbf{1236}][\mathbf{1456}] = 0, \\[2ex] [\mathbf{m}_1\mathbf{m}_2\mathbf{m}_3][\mathbf{m}_2\mathbf{m}_4\mathbf{m}_5][\mathbf{1246}][\mathbf{2356}] \\ -[\mathbf{m}_1\mathbf{m}_2\mathbf{m}_4][\mathbf{m}_2\mathbf{m}_3\mathbf{m}_5][\mathbf{1236}][\mathbf{2456}] = 0. \end{cases} \tag{4}$$

After a permutation of $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$ and the corresponding image points, this equation system is still the invariant relationship of $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{O}$ lying on the same twisted cubic, but is not independent of the above one.

Proof. Proposition 1 can be obtained by (3) and (2) as follows. If $\mathbf{O}$ lies on the twisted cubic through $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$, then $\mathbf{O}$ satisfies (3), further by (2), we have:

$$\begin{cases} \dfrac{[\mathbf{1246}][\mathbf{1356}]}{[\mathbf{1236}][\mathbf{1456}]} = \dfrac{[\mathbf{124O}][\mathbf{135O}]}{[\mathbf{123O}][\mathbf{145O}]} = \dfrac{[\mathbf{m}_1\mathbf{m}_2\mathbf{m}_4][\mathbf{m}_1\mathbf{m}_3\mathbf{m}_5]}{[\mathbf{m}_1\mathbf{m}_2\mathbf{m}_3][\mathbf{m}_1\mathbf{m}_4\mathbf{m}_5]}, \\[2ex] \dfrac{[\mathbf{1246}][\mathbf{2356}]}{[\mathbf{1236}][\mathbf{2456}]} = \dfrac{[\mathbf{124O}][\mathbf{235O}]}{[\mathbf{123O}][\mathbf{245O}]} = \dfrac{[\mathbf{m}_1\mathbf{m}_2\mathbf{m}_4][\mathbf{m}_2\mathbf{m}_3\mathbf{m}_5]}{[\mathbf{m}_1\mathbf{m}_2\mathbf{m}_3][\mathbf{m}_2\mathbf{m}_4\mathbf{m}_5]}. \end{cases} \quad (5)$$

Notice that in (3), there is another condition such as: **X** is not on the line through **1** and **2**. Here for the optical center **O**, this additional condition is unnecessary because if **O** is on the line through **1** and **2**, then $\mathbf{m}_1$ and $\mathbf{m}_2$ become one point, which is contrary to our assumption that no three image points are collinear.

Because the representation (3) is independent of the order of $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$, the invariant relationship (4) is also independent of the order of them and their corresponding image points. Proposition 1 is proved.

Each ratio in (5) is a cross ratio [5]. Therefore, (4) is an invariant relationship between space points and image points equivalent to (5).

By the last paragraph of Section 2 or [14], we know the image points $\mathbf{m}_i$, $i =$ 1..6, from the twisted cubic configuration are con-conic. But, this is not a sufficient condition for this configuration [14].

We have established the above invariant relationship between six space points and their images under a perspective view when the optical center and the space points lie on a twisted cubic. The number of the equations describing the relationship is two, which is different from the number one for the previous general configuration [2, 3, 4, 7, 9, 10, 11, 12, 13]. The reason is just from the degeneracy of the twisted cubic configuration [17].

This invariant relationship can be easily extended to invariant relationship between two perspective views that the two camera optical centers and six space points lie on a twisted cubic. Then the result can be used to detect critical data for computing fundamental matrix or epipoles [6, 8].

3.2 Establishing an Algorithm to Detect Twisted Cubic Degenerate Configuration of Six Points

Detecting the twisted cubic degenerate configuration is important because the data for camera parameter estimation from the degenerate configuration is critical and can result in useless recovered camera parameters.

We apply the established invariant relationship to detect the twisted cubic degenerate configuration of six points. The method is based on a criterion function that can be faithfully trusted for camera parameter estimation from six points in practice.

Notice that in the invariant relationship (4) of the twisted cubic case, $\mathbf{m}_6$ does not occur. And, by the last paragraph of Section 2, we know that the first equation of (4) is the cone with **1** as the vertex, the second equation of (4) is the cone with **2** as the vertex. Thus, the two equations are denoted as $g_{1,(24,35)} = 0$, $g_{2,(14,35)} = 0$, which can be criterion functions to recognize the twisted cubic degenerate configuration. But, stability to noise is much affected by the order of space points and image points. So, we are to consider more equations after changing the orders of space points and their corresponding images. We do a permutation on $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$ and their corresponding images in $g_{1,(24,35)}$ or $g_{2,(14,35)}$ and denote the corresponding result as $g_{i,(jk,pq)}$. For each of such permuted functions, we also assign a weigh to it, and then the criterion function on $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$ and their image points is constructed as:

$$f = \frac{1}{90} \sum_{i=1}^{6} \sum_{\sigma \in S} \frac{1}{W_{i,\sigma}^2} g_{i,\sigma}^2,$$

where $S = \{(jk,pq), (jp,kq), (jq,kp), (jk,pl), (jp,kl), (jl,kp), (jk,ql), (jq,kl), (jl,kq), (jp,ql), (jq,pl), (jl,pq), (kp,ql), (kq,pl), (kl,pq)\}$ has 15 elements and $\{i,j,k,p,q,l\} = \{1,2,3,4,5,6\}$. According to our experience from extensive experiments, the weigh $W_{i,\sigma}$ is taken as the mean of the absolute values of the two terms in $g_{i,\sigma}$ in this work.

Now, we can propose a two-step algorithm to determine whether six space points and the camera optical center lie on a proper twisted cubic or not, where no four of the space points are coplanar, and no three of the image points are collinear.

Step 1. Compute the value of the criterion function f on the six space points $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}$, $\mathbf{5}, \mathbf{6}$ and their corresponding image points.

Step 2. Let ϵ be a preset threshold, and determine whether $f < \epsilon$. If yes, then $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$ and the camera optical center lie on a proper twisted cubic. Otherwise, they are not on a twisted cubic.

It is clear that the criterion function f is only on the image and space points, and in it there is no any computation on the camera optical center or projective matrix. f can let us efficiently know whether space points and camera lie on the same twisted cubic or not, also it can be faithfully trusted for camera parameter estimation from six points in practice as shown in real experiments. According to our experience from extensive experiments, the threshold ϵ is taken as 1.1 in this work.

4 Experiments

4.1 Simulations

We perform experiments on simulated data to test the stability of the proposed algorithm in the following. The world coordinate system is taken as the camera coordinate system. The simulated camera intrinsic parameters are:

$$\mathbf{K} = \begin{pmatrix} 1000 & 0 & 512 \\ 0 & 900 & 384 \\ 0 & 0 & 1 \end{pmatrix},$$

then we generate the images of seven space points $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}$, $\mathbf{5}, \mathbf{6}, \mathbf{7}$ such that $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{O}$ do not lie on a twisted cubic, and $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{7}, \mathbf{O}$ do lie on a twisted cubic. The Gaussian noise with mean 0 and standard deviation ranging from 0 to 6 pixels is directly added to each image points, and then I_1: the value of the criterion function of f on $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}$ and their image points and I_2: the value of the criterion function of f on $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{7}$ and their image points are computed. For each noise level, we perform 100 runs, and the averaged results are calculated, still denoted by I_1, I_2. Since $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{7}, \mathbf{O}$ lie on a twisted cubic and $\mathbf{1}$, $\mathbf{2}$, $\mathbf{3}$, $\mathbf{4}$, $\mathbf{5}$, $\mathbf{6}$, $\mathbf{O}$ do not lie on a twisted cubic, I_2 should be close to zero, while I_1 should not. And therefore there should be $I_1 > I_2$.

We do the repeated simulations, the image sizes are not greater than 1000 × 1000 pixels. We find that I_1 and I_2 are all very stable, and there are always $I_1 > 1.1 > I_2$. Some image data D_i, $i = 1..8$ are shown in Fig. 1, where the images of $\mathbf{m}_i$, $i = 1..5$ in D_i, $i = 1..6$ distribute rather evenly, and the ones in D_7, D_8 do not ($\mathbf{m}_2, \mathbf{m}_5$ are very close). The corresponding results of I_1, I_2 of them are shown in Table 1.

Though there is noise, the variations of I_1, I_2 are very small, and there are always $I_1 > I_2$. These show that the proposed algorithm can distinguish robustly between the twisted cubic degenerate configuration and the nondegenerate configuration, and the criterion function f is quite stable against noise.

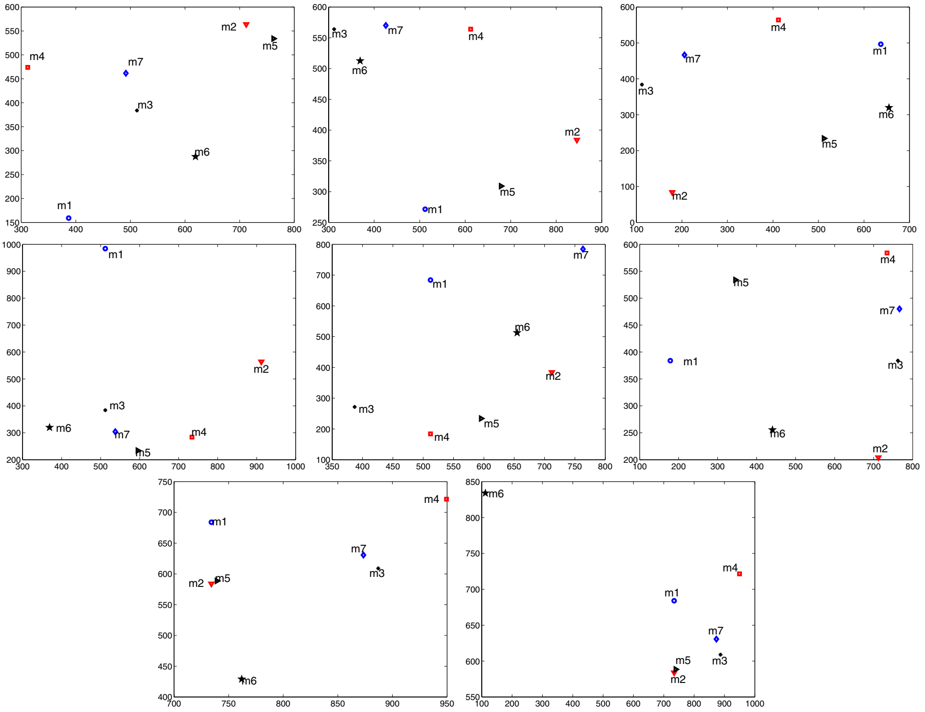


Fig. 1. Some image data, denoted as D_i, $i = 1..8$, where $\mathbf{m}_2$ and $\mathbf{m}_5$ are very close in D_7 and D_8

Table 1. The values I_1, I_2 under different noise levels

Noise level (pixel)		0	0.5	1	1.5	2
D_1	I_1	2.41	2.41	2.41	2.41	2.41
	I_2	0.00	0.00	0.00	0.00	0.00
D_2	I_1	2.94	2.94	2.94	2.93	2.94
	I_2	0.00	0.00	0.01	0.03	0.05
D_3	I_1	2.67	2.67	2.67	2.67	2.67
	I_2	0.00	0.00	0.00	0.00	0.00
D_4	I_1	2.19	2.18	2.18	2.17	2.17
	I_2	0.00	0.01	0.00	0.01	0.01
D_5	I_1	2.86	2.86	2.86	2.86	2.86
	I_2	0.00	0.00	0.00	0.00	0.00
D_6	I_1	2.25	2.25	2.25	2.25	2.25
	I_2	0.00	0.00	0.00	0.01	0.01
D_7	I_1	2.75	2.77	2.76	2.77	2.75
	I_2	0.00	0.11	0.25	0.48	0.52
D_8	I_1	2.55	2.55	2.52	2.48	2.46
	I_2	0.00	0.14	0.30	0.54	0.64

4.2 Experiments from Real Data

This section will show the usefulness of the proposed criterion function f that can be faithfully trusted for camera parameter estimation from six points in practice.

We are to calibrate a camera from a single view of a grid. We choose two groups of six pairs of space and image points with larger value and smaller value of the criterion function f, then from them, estimate camera parameters and compare the results.

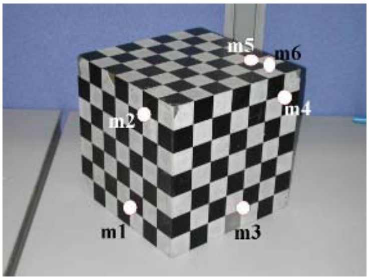


Fig. 2. A real image of a calibration grid, where $\mathbf{m}_i, \mathbf{i} = 1..6$ are the image points from G_{min} and G_{min} is the group of six pairs of space and image points with the minimal value of the criterion function f

The used image taken by a CCD camera is shown in Fig. 2. The size of the image is of 1024 × 768 pixels. We extract the pixels of the edges by Canny edge detector, then fit them as lines, and calculate the intersection points of these lines. The world coordinate system is set up in the grid. Then we have 108 pairs of space points and the corresponding image points. By using DLT method [1] from these 108 pairs of space and image points, we obtain the camera intrinsic parameter matrix $\mathbf{K}$ and the camera pose parameters: rotation $\mathbf{R}$ and translation $\mathbf{t}$ as follows:

$$\mathbf{K} = \begin{pmatrix} 2049.8128 & -2.7983 & 523.9202 \\ 0 & 2050.5605 & 394.1385 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{R} = \begin{pmatrix} 0.7784 & -0.6272 & 0.0270 \\ -0.2648 & -0.3671 & -0.8917 \\ 0.5692 & 0.6870 & -0.4518 \end{pmatrix},$$
$$\mathbf{t} = \left(-0.7503, 4.8624, 30.7296\right)^T .$$

From these 108 pairs of space and image points, we randomly combine 125 groups of six pairs with no three image points collinear, no three space points collinear, and no four space points coplanar. We choose the group with the maximal value of the criterion function f, the group with the minimal value of the criterion function f, and denote them as G_{max}, G_{min} respectively. The value of f from G_{max} is 3.0997, and the value of f from G_{min} is 0.0380.

The image points of G_{min} are plotted as $\mathbf{m}_i,\ i = 1..6$ as shown in Fig. 2. We can see that there is no linear relation among them. The value of f, 0.0380, says that they are from degenerate configuration.

We calibrate the camera from the six pairs of space and image points in G_{max} by DLT method, and the results are:

$$\mathbf{K}_1 = \begin{pmatrix} 2140.9987 & -2.1069 & 570.3262 \\ 0 & 2138.6413 & 452.6338 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{R}_1 = \begin{pmatrix} 0.7668 & -0.6409 & 0.0358 \\ -0.2772 & -0.3808 & -0.8822 \\ 0.5790 & 0.6665 & -0.4696 \end{pmatrix},$$
$$\mathbf{t}_1 = \left(-1.4424, 3.9693, 32.1664\right)^T .$$

Similarly, we calibrate the camera from the six pairs of space and image points in G_{min}, and the results are:

$$\mathbf{K}_2 = \begin{pmatrix} 980.4078 & 26.8782 & 430.9372 \\ 0 & 870.5114 & 541.8497 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{R}_2 = \begin{pmatrix} -0.6666 & 0.7454 & 0.0062 \\ -0.0205 & -0.0266 & 0.9994 \\ 0.7451 & 0.6661 & 0.0330 \end{pmatrix},$$
$$\mathbf{t}_2 = \left(-1.0456, -1.5375, -18.0317\right)^T .$$

We evaluate the estimated $\mathbf{K}_1$ and $\mathbf{K}_2$ by comparing them with $\mathbf{K}$:

$$\mathbf{K}_1 - \mathbf{K} = \begin{pmatrix} 91.1859 & 0.6914 & 46.4060 \\ 0 & 88.0809 & 58.4953 \\ 0 & 0 & 0 \end{pmatrix},$$
$$\mathbf{K}_2 - \mathbf{K} = \begin{pmatrix} -1069.4050 & 29.6765 & -92.9830 \\ 0 & -1180.0491 & 147.7112 \\ 0 & 0 & 0 \end{pmatrix}.$$

It is clear that the absolute error for each intrinsic parameter from $\mathbf{K}_2$ is much greater than each one from $\mathbf{K}_1$. We evaluate the recovered $\mathbf{R}_1$ and $\mathbf{R}_2$ by comparing them with $\mathbf{R}$:

$$\mathbf{R}_1-\mathbf{R}=\begin{pmatrix} -0.0116 & -0.0137 & 0.0087 \\ -0.0123 & -0.0137 & 0.0095 \\ 0.0098 & -0.0205 & -0.0178 \end{pmatrix}, \mathbf{R}_2-\mathbf{R}=\begin{pmatrix} -1.4450 & 1.3725 & -0.0208 \\ 0.2443 & 0.3405 & 1.8911 \\ 0.1759 & -0.0208 & 0.4848 \end{pmatrix}$$

And also, the recovered $\mathbf{t}_1$ and $\mathbf{t}_2$ are evaluated by the differences between them and $\mathbf{t}$:

$$\mathbf{t}_1-\mathbf{t}=\left(-0.6922,-0.8931,1.4368\right)^T, \mathbf{t}_2-\mathbf{t}=\left(-0.2954,-6.3999,-48.7613\right)^T.$$

Also, it is clear that the accuracies of $\mathbf{R}_1, \mathbf{t}_1$ are higher than the accuracies of $\mathbf{R}_2, \mathbf{t}_2$ except the first element of the translation. For the first element of the translation, the absolute error from $\mathbf{t}_1$ is greater than the absolute error from $\mathbf{t}_2$, but the difference between these two absolute errors is not so large as that for the second or third element of the translation.

So, we can see that the calibration result from six pairs of space and image points with smaller value of the criterion function f (i.e. space points and optical center are near to the twisted cubic degenerate configuration) is not better than the one from six pairs of space and image points with larger value of the criterion function f (i.e. space points and optical center are far from the twisted cubic degenerate configuration). The proposed criterion function f, thus, can be faithfully trusted for camera parameter estimation from six points. We also perform the experiments from other real images and obtain the similar results. The details are omitted due to the space limit.

5 Summary and Conclusions

We establish the invariant relationship between six space points and their images under a perspective view when camera optical center and the space points lie on a twisted cubic. Then, the invariant relationship is used to recognize the nontrivial degenerate configuration of six points through a new algorithm. The algorithm is based on a criterion function, does not need explicit computations on the optical center or projective matrix, and is shown stable and robust against noise. We believe that it has further usefulness. For example, when applying RANSAC during the process of determining camera parameters, the critical groups of data can be filtered by the criterion function of this method, and then the algorithm can be extended to more than six pairs of space points and image points. The sample of six pairs of space points and image points with poor performance will not be chosen in RANSAC. How to know whether a sample is unreliable or not? The criterion function in this paper just can be used to detect the unreliability. We will report this work in future. The invariant relationship can also be easily extended to the the invariant relationship between two perspective views when the two camera optical centers and space points lie on the same twisted cubic, and then the result can similarly be used to detect critical data for computing fundamental matrix or epipoles [6, 8].

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant No. 60303021 and No. 60475009.

References

1. Abdel-Aziz, Y.I., Karara, H.M.: Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-range Photogrammetry. *Proc. ASP/UI Symp. on Close Range Photogrammetry* (1971)1-18.
2. Bayro-Corrochano, E., Banarer, V.: A Geometric Approach for the Theory and Applications of 3D Projective Invariants. *Journal of Mathematical Imaging and Vision*, Vol. 16, No. 2 (2001)131-154.
3. Carlsson, S.: Symmetry in Perspective. *ECCV* (1998)249-263.
4. Carlsson, S.: View Variation and Linear Invariants in 2-D and 3-D. Tech. Rep. ISRN KTH/NA/P–95/22–SE, Dec. (1995).
5. Carlsson, S.: The Double Algebra: An Effective Tool for Computing Invariants in Computer Vision. *Applications of Invariance in Computer Vision: Joint European–US Workshop*, Azores, Portugal, Oct. (1993)145-164.
6. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000).
7. Hartley, R.: Projective Reconstruction and Invariants from Multiple Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 16, No. 10 (1994)1036-1041.
8. Maybank, S.: Theory of Reconstruction from Image Motion. Springer-Verlag (1993).
9. Maybank, S.: Relation between 3D Invariants and 2D Invariants. *IEEE Workshop on Representation of Visual Scenes*, Cambridge, Massachusetts, Jun. (1995).
10. Quan, L.: Invariants of 6 Points from 3 Uncalibrated Images. *ECCV* (1994)459-470.
11. Quan, L.: Invariants of Six Points and Projective Reconstruction from Three Uncalibrated Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 17, No. 1 (1995)34-46.
12. Roh, K.S., Kweon, I.S.: 3-D Object Recognition Using a New Invariant Relationship by Single-view. *Pattern Recognition*, Vol. 33, No. 5 (2000)741-754.
13. Schaffalitzky, F., Zisserman, A., Hartley, R., Torr, P.: A Six Point Solution for Structure and Motion. *ECCV* (2000)632-648.
14. Semple, J., Kneebone, G.: Algebraic Projective Geometry. Oxford University Press (1952).
15. Weng, J., Huang, T.S., Ahuja, N.: Motion and Structure from Two Perspective Views: Algorithms, Error Analysis, and Error Estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 5 (1989)451-476.
16. Wu, Y.H., Hu, Z.Y.: The Invariant Representations of a Quadric Cone and a Twisted Cubic. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 25, No. 10 (2003)1329-1332.
17. Wu, Y.H., Hu, Z.Y.: A Unified and Complete Framework of Invariance for Six Points. In: Li, H., Olver, P.J., and Sommer, G. (Eds.), *Computer Algebra and Geometric Algebra with Applications, IWMM/GIAE* 2004, Springer-Verlag, LNCS 3519 (2005)403-417.

Robustness in Motion Averaging

Venu Madhav Govindu

HIG-25, Simhapuri Layout,
Visakhapatnam, AP, 530047 India
venu@narmada.org

Abstract. The averaging of multiple pairwise relative motions in a sequence provides a fast and accurate method of camera motion estimation with a wide range of applications, including view registration, robotic path estimation, super-resolution. Since this approach involves averaging in the Lie-algebra of the underlying motion representation, it is non-robust and susceptible to contamination due to outliers in the individual relative motions. In this paper, we introduce a graph-based sampling scheme that efficiently remove such motion outliers. The resulting global motion solution is robust and also provides an empirical estimate of the inherent statistical uncertainty. Example results are provided to demonstrate the efficacy of our approach to incorporating robustness in motion averaging.

1 Introduction

Estimation of the camera motion from an image sequence is a well-studied problem [1]. Most conventional approaches can be classified either as *algebraic* methods involving a few frames or *optimisation* methods that solve for the global motion of the entire sequence. Examples of the former approach include epipolar and trilinear geometry representations whereas motion estimation using bundle-adjustment is an example of the later approach. While the algebraic approaches are fast, they are inherently inaccurate as they use information from only a few frames. In contrast, bundle-adjustment results in accurate solutions but is computationally expensive and also requires an accurate initial guess.

To overcome both these limitations, averaging of relative motions between image pairs was introduced in [2] and further developed in [3]. In this approach the efficiency of the algebraic approach was exploited to provide multiple relative motion estimates between image pairs that were subsequently averaged resulting in a fast, flexible and accurate estimate of the global motion. This method uses the Lie group structure of the motion representation to give a principled algorithm for averaging of relative motions. The averaging scheme has a wide range of applications including camera motion estimation, robotic path reconstruction, multi-view registration, and super-resolution. However since the method involves averaging of multiple relative motions in their corresponding Lie-algebra, it is inherently susceptible to error due to contamination by outliers. This is the property of any scheme that involves averaging of multiple observations, for example the arithmetic average of a scalar $\hat{\mathbf{x}} = \frac{1}{N}\sum \mathbf{x}_i$. A single outlier element $\mathbf{x}_i$ will cause the estimate $\hat{\mathbf{x}}$ to be grossly incorrect. In the case of relative motions, the outliers may arise due to incorrect feature correspondences. In this paper we introduce a randomised sampling scheme that can detect such outliers in the

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 457–466, 2006.

set of relative motions estimated from a sequence. While we will elaborate the approach in subsequent sections, we briefly describe our approach here. In the spirit of the RANSAC [4, 5] approach to robustness we derive global motion estimates that involve the minimal number of pairwise observations. As we shall show in Sec. 3 this is equivalent to selecting minimum spanning trees (MST) of a graph. The relative motions that survive the sampling process are data *inliers* that can be averaged resulting in accurate and robust estimates. The sampling scheme can be further applied to the inliers themselves to provide covariance estimates which is equivalent to the bootstrap approach to empirical estimation of uncertainty [6].

The rest of the paper is organised as follows. In Sec. 2 we describe the motion averaging scheme presented. Sec. 3 motivates and develops the graph-sampling based approach to outlier detection in relative motions. The result of applying this approach to a real image sequence is shown in Sec. 4. Finally, Sec. 5 presents some conclusions and directions for further work.

2 Averaging of Relative Motions

In this section we summarise previous results on motion averaging. The following analysis applies equally to both rotation and Euclidean motion estimation and a linear solution for this formulation was described earlier in [2] and developed into a Lie group representation in [3]. For N images, the globally motion can be described by $N-1$ motions, if we pick any image as the reference frame. Without loss of generality, we can assume that the reference frame is attached to the first image frame. We denote the motion between frame i and the reference frame as $\mathbf{M}_i$, and the relative motion between two frames i and j as $\mathbf{M}_{ij}$, where $\mathbf{M}_{ij} = \mathbf{M}_j{\mathbf{M}_i}^{-1}$. This relationship captures the notion of "consistency", i.e. the composition of any series of transformations starting from frame i and ending in frame j should be identical to $\mathbf{M}_{ij}$ (See Fig. 1). Due to the presence of noise in our observations the various transformation estimates would not be consistent with each other. Hence $\mathbf{M}_{ij} \neq \mathbf{M}_j{\mathbf{M}_i}^{-1}$, where $\mathbf{M}_{ij}$ is the estimated transformation between frames i and j. However we can rewrite the given relationship as a constraint on the global motion model $\{\mathbf{M}_2, \cdots, \mathbf{M}_N\}$ which completely describes the motion. The first image being the reference frame, $\mathbf{M}_1$ is an identity transformation.

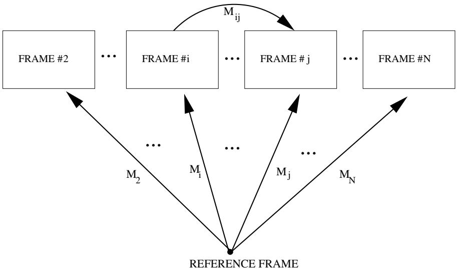


Fig. 1. The relative motions are estimated from the data. The global motion with respect to the first frame is estimated by averaging the over-determined set of relative motion constraints.

Since in general we have upto $\frac{N(N-1)}{2}$ such constraints, we have an over-determined system of equations.

$$\mathbf{M}_j\mathbf{M}_i{}^{-1} = \mathbf{M}_{ij}, \forall i \neq j \tag{1}$$

where the variables on the left-side are unknowns to be estimated ("fitted") in terms of the observed data $\mathbf{M}_{ij}$ on the right. Intuitively, we want to estimate a global motion model $\{\mathbf{M}_i\}$ that is most consistent with the measurements $\{\mathbf{M}_{ij}\}$ derived from the data. Thus the errors in individual estimates of $\mathbf{M}_{ij}$ are "averaged" out resulting in reduced error. It may be noted that in Eqn. 1, we are not required to use every pairwise constraint. For extended sequences, there is seldom any overlap between frames well separated in time, therefore their relative two-frame motions cannot be estimated. However we can still get a consistent solution as long as we have at least $N-1$ relative motions. In fact, the sampling procedure to be outlined in Sec. 3 exploits this property to incorporate robustness into the estimation procedure.

2.1 Averages on the Lie Group

The idea of averaging on the Lie group is at the heart of the motion averaging approach used in this paper. In this subsection we shall provide an extremely elementary summary of the properties of Lie groups and the related approach to averaging. For further details, the reader should consult [3]. A group G is a set whose elements satisfy the relationships of *associativity*, *identity* and the existence of an *inverse*. A Lie group is a group which also behaves like a smooth, differentiable manifold. Intuitively, Lie groups can be locally viewed as topologically equivalent to the vector space, $\mathbb{R}^n$ and can be locally described by its tangent-space whose elements form a Lie algebra $\mathfrak{g}$. The Lie algebra $\mathfrak{g}$ is equipped with a bilinear operation $[.,.] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$ known as the Lie bracket which satisfies the property of *anti-symmetry* and the *Jacobi identity*. All finite-dimensional Lie groups have matrix representations and the bracket in this case is the commutator operation $[\mathbf{x}, \mathbf{y}] = \mathbf{xy} - \mathbf{yx}$. The Lie algebra and the associated Lie group are related by the exponential mapping. This exponential mapping and its inverse (i.e. logarithm) enable us to freely operate in either the Lie group or its associated algebra according to convenience. The motion models that we are interested in, namely three-dimensional rotations and three-dimensional Euclidean motion are elements of the Special Orthogonal $\mathbf{SO(3)}$ and Special Euclidean $\mathbf{SE(3)}$ groups respectively. For non-commutative Lie groups, the usual exponential relation $e^{\mathbf{x}}e^{\mathbf{y}} = e^{\mathbf{x}+\mathbf{y}}$ does not hold. The equivalent mapping is defined by $d : \mathfrak{g} \times \mathfrak{g} \mapsto \mathfrak{g}$, i.e. $e^{\mathbf{x}}e^{\mathbf{y}} = e^{d(\mathbf{x},\mathbf{y})}$, where $d(.,.)$ is given by the Baker-Campbell-Hausdorff (BCH) formula [7] and is the intrinsic (Riemannian) distance on the manifold representing the group. For example, for rotations $\boldsymbol{\omega_1}$ and $\boldsymbol{\omega_2}$, $d(\boldsymbol{\omega_2}, -\boldsymbol{\omega_1})$ represents the rotation ("distance") that will take us from $\boldsymbol{\omega_1}$ to $\boldsymbol{\omega_2}$. Using this *intrinsic* distance between points on a Riemannian manifold the 'intrinsic' average can be defined as

$$\mu = \arg \min_{\mathbf{X} \in G} \sum_{k=1}^{N} d^2(\mathbf{X}_k, \mathbf{X})$$

In general this intrinsic average is preferable to other approximations as the estimation process always confirms to the underlying group structure involved. The reader can refer to [8] for further details including an algorithm for averaging elements on the group manifold. For matrix groups, the Riemannian distance is defined by the matrix logarithm operation. By using the BCH formula this distance can be approximated as $d(\mathbf{X}, \mathbf{Y}) = ||\log(\mathbf{Y}\mathbf{X}^{-1})|| \approx ||\log(\mathbf{Y}) - \log(\mathbf{X})|| = ||\mathbf{y} - \mathbf{x}||$ where $\mathbf{x}$ and $\mathbf{y}$ are logarithms of matrices $\mathbf{X}$ and $\mathbf{Y}$ respectively.

2.2 Lie Averaging of Relative Motions

The scheme for averaging relative motions is similar in spirit to the intrinsic averaging approach of [8]. Starting from the constraint $\mathbf{M}_{ij} = \mathbf{M}_j\mathbf{M}_i{}^{-1}$, by applying the first-order approximation to the Riemannian distance, we have $\mathfrak{m}_{ij} = \mathfrak{m}_j - \mathfrak{m}_i$ since $\mathfrak{m} = \log(\mathbf{M})$. Arranging in the form of a column vector, we have $\mathfrak{v} = vec(\mathfrak{m})$ implying $\mathfrak{v}_{ij} = \mathfrak{v}_j - \mathfrak{v}_i$. If we stack all the column vectors for the global motion model into one big vector $\mathfrak{V}$ we have $\mathfrak{V} = [\mathfrak{v}_2; \cdots; \mathfrak{v}_N]$. Given this unified vector representation for the global motion model, we have

$$\begin{aligned} \mathbf{M}_{ij} = \mathbf{M}_j\mathbf{M}_i{}^{-1} &\Rightarrow \mathfrak{m}_{ij} = \mathfrak{m}_j - \mathfrak{m}_i \\ \Rightarrow \mathfrak{v}_{ij} = \mathfrak{v}_j - \mathfrak{v}_i &= \underbrace{\left[\cdots - \mathbf{I} \cdots \mathbf{I} \cdots\right]}_{=\mathbf{D}_{ij}} \mathfrak{V} \end{aligned} \tag{2}$$

where $\mathbf{I}$ denotes an identity matrix. While Eqn. 2 denotes a single relative motion in terms of the global motion model, we can stack all the relative motion vectors $\mathfrak{v}_{ij}$ into one big vector $\mathbb{V}_{ij} = [\mathfrak{v}_{ij1}; \mathfrak{v}_{ij2}; \cdots]$ where $ij1, ij2$ etc. denote different relative motion indices. Similarly we can stacked $\mathbf{D} = [\mathbf{D}_{ij1}; \mathbf{D}_{ij2}; \cdots]$ leading to

$$\begin{aligned} \mathbf{M}_j\mathbf{M}_i{}^{-1} &= \mathbf{M}_{ij} \\ \rightsquigarrow \mathbf{D}\mathfrak{V} &= \mathbb{V}_{ij} \Rightarrow \mathfrak{V} = \mathbf{D}^{\dagger}\mathbb{V}_{ij} \end{aligned} \tag{3}$$

where $\mathbf{D}^{\dagger}$ is the pseudo-inverse. This results in the following iterative scheme :

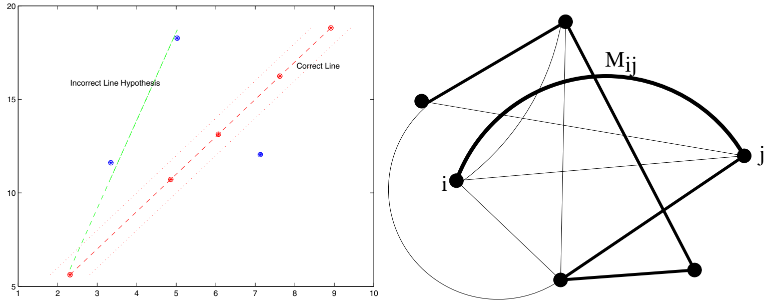
A1 : Algorithm for Relative Motion Averaging

```
Input : ${\mathbf{M}_{ij1}, \mathbf{M}_{ij2} \cdots, \mathbf{M}_{ijn}}$ ($n$ relative motions)
Output : $\mathbf{M}_g : {\mathbf{M}_2, \cdots, \mathbf{M}_N}$ ($N$ image global motion)
Set $\mathbf{M}_g$ to an initial guess (Linear solution in [2])
$Do$
$\Delta\mathbf{M}_{ij} = \mathbf{M}_j{}^{-1}\mathbf{M}_{ij}\mathbf{M}_i$
$\Delta\mathfrak{m}_{ij} = log(\Delta\mathbf{M}_{ij})$
$\Delta\mathfrak{v}_{ij} = vec(\mathfrak{m}_{ij})$
$\Delta\mathfrak{V} = \mathbf{D}^{\dagger}\Delta\mathbb{V}_{ij}$
$\forall k \in [2, N], \mathbf{M}_k = \mathbf{M}_k exp(\Delta\mathfrak{v}_k)$
$Repeat\ till\ ||\Delta\mathfrak{V}|| < \epsilon$
```

While further details cannot be provided here due to space constraints, for our purposes it is sufficient to note that we can use the above approach to accurately average the relative motions on the appropriate Lie group representation.

3 Sampling on the View-Graph of Relative Motions

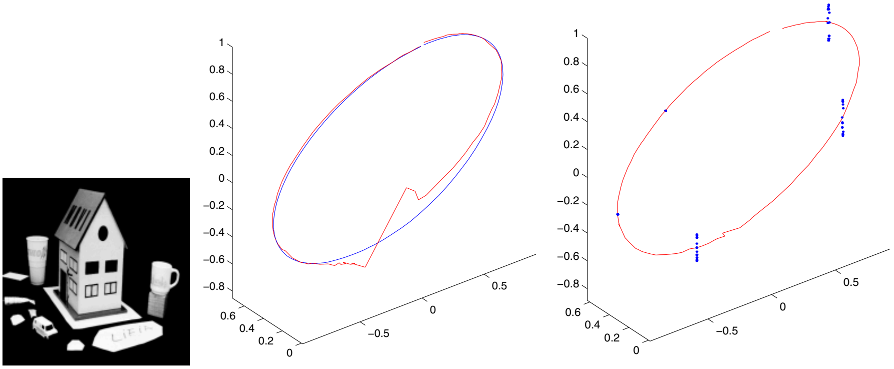
While the algorithm described in Sec. 2 is an effective scheme for estimating the global camera motion from multiple estimates of relative motions, it suffers from the limitation of being non-robust. Consider a scenario where an individual relative motion is corrupted, say, due to incorrect correspondences used in the estimation of epipolar geometry. This would result in an incorrect estimate for $\mathbf{M}_{ij}$. When this incorrect measurement is incorporated into the averaging scheme of Algorithm **A1**, the entire result would be corrupted. Therefore, we require a procedure that would identify outliers in the set of relative motions and discard them prior to the averaging of these measurements using the Lie-algebraic averaging scheme.



(a) RANSAC Example (b) Viewgraph of relative motions

Fig. 2. (a) illustrates the RANSAC approach. Some points fall on a line whereas others are outliers. (b) shows a view graph representing relative motions identified by the vertices. Each edge represents an estimated relative motion between the two vertices. The bold edges represent a minimum spanning tree (MST).

A well-known approach for incorporating robustness in computer vision is the Randomised Sampling Consensus (RANSAC) method [4, 5]. This randomised scheme has been shown to have desired statistical properties in that it can effectively identify data outliers that do not satisfy a given geometric model. The idea behind RANSAC is illustrated in Fig. 3(a) where we have points that lie on a straight line along with some outlier points. If we were to seek the least squares fit for the full set of data points the resulting line solution would be grossly incorrect as it would average over the correct points and the outliers. The RANSAC approach to detecting outliers works by generating solutions that use the *minimal* number of data points. Since a line can be defined by two non-identical points, we randomly select a pair of points and use the line passing through them as our *hypothesis*. All points that fall within a pre-specified distance (say D) from this hypothesis line are declared to fit the line. In Fig. 3(a) this range is indicated by the two dotted lines around the true line. For each trial, we count the number of points that fall within this bounding region. For a given number of trials, the hypothesis with the maximum number of points within the bounding region is selected and all points within the bounding region are declared as



(a) MOVI data (b) Non-robust Averaging (c) Robust Averaging

Fig. 3. (a) shows one image from the MOVI sequence; (b) shows the results from [9] and the incorrect estimate due to outliers; (c) shows our estimate after automatic removal of the outliers. The covariance is shown in an exaggerated form for visualisation. See text for details.

inliers and those outside this region are classified as outliers. The line estimate is now obtained by least-squares fitting of all inliers. The green line in Fig. 3(a) indicates a line hypothesis that includes outliers, but the score for this line will always be less than that for a true hypothesis, implying robustness to as many as 50% of data outliers.

3.1 A Robust Algorithm for Motion Averaging

In the case of motion models that describe the global motion we can develop a sampling method similar in spirit to RANSAC. We can describe the information of all the relative motions estimated in a sequence in a graph. Consider a graph $G = (V, E)$ where V is the set of vertices and E the set of edges. Each vertex of the graph denotes an individual image, resulting in N vertices. If we are able to estimate the relative motion between a pair of images i and j we add an edge E_{ij} between the said vertices. Such a representation of relative motions is called a view-graph and capture all the information available and has also been used to solve other problems, for instance see [10]. We show an example of such a view-graph in Fig. 3(b). The absence of an edge connecting two vertices implies that the relative motion between those two vertices in not available. To keep our analysis simple, we assume that we are given a set of relative motions $\{\mathbf{M}_{ij}\}$ and no more information to indicate their reliability. Therefore no weight information is used for the edges, i.e. all edges have the same weight. Moreover the resulting graph is bidirectional.

Since the RANSAC approach requires a *minimal* solution we need such a solution that can capture the global motion for the image sequence. Since the relative motions between images are represented by edges on the view graph it will be immediate obvious that the minimal solution for our problem is given by the *minimum spanning tree* (MST) of the graph G. When the graph G has a single connected component, the minimum spanning tree is a set of edges such that every vertex in V is reachable from every other vertex in V and the total weight of all edges in the tree is minimum. For a graph with N vertices, the minimum spanning tree always has $N-1$ edges [11]. In Fig. 3(b) we have a graph representing the relative motions available and an MST is shown in bold edges. Since in an MST every vertex is reachable from any vertex, given an MST and the corresponding relative motions we can solve for the global motion model[1]. Now given an MST on the view-graph G, we can solve for the global motion model $\{\mathbf{M}_i\}$ and consequently every relative motion can be compared to this solution. For example, if the global motion model for an MST is $\mathbf{M} = \{\mathbf{M}_2, \cdots, \mathbf{M}_N\}$ and the relative motion between vertices i and j is given by $\mathbf{M}_{ij}$, then the "distance" of this edge from the global motion model is given by $d(\mathbf{M}_{ij}, \mathbf{M}_j{\mathbf{M}_i}^{-1})$.

Each MST of the view-graph represents a model hypothesis, i.e. a solution for the global motion. Given a pre-specified distance threshold, we can count the number of relative motions (i.e. edges) that fall within this distance from the global motion. Thus for each MST, we count the number of *inliers* in the original set of relative motions. This is repeated for a given number of trials and the MST with the maximum number of inliers declared the winner. Subsequently we use Algorithm **A1** to solve for the global motion using all inliers, resulting in an accurate solution that is also robust to the presence of outliers. Since each edge has the same weight and every MST has $N-1$ edges, the total weight for all spanning trees is the same. Therefore for our problem we need to generate many spanning trees for the view-graph. This can be achieved by randomising a *depth first search* (DFS) on the graph G. The DFS is a standard algorithm for systematically creating a tree given a starting vertex of a graph. In our modification, in each instance we start at a random vertex and at every parent vertex, we randomly pick the next adjacent vertex to be visited in the search process. For each run of this procedure we generate a spanning tree that is used in the RANSAC procedure as described above.

While in this paper we have chosen to ascribe equal weights to all edges, in the presence of appropriate measures of reliability for each individual relative motion estimate, we can easily incorporate that information as a weight on the view-graph G. For example, if $\mathbf{e}_{ij}$ is the root mean squares error for the estimation procedure for relative motion $\mathbf{M}_{ij}$ we can choose the weight for the edge connecting vertices i and j as $\mathbf{w}_{ij} = {\mathbf{e}_{ij}}^2$. In such a scenario the minimum spanning tree procedure will seek to minimise the sum of the edge weights, which is equivalent to a minimal solution for the global motion model with the least squared error for all the measurements used. However since now the edge weights are not identical the procedure for generating a randomised MST has to utilise the weight information. The algorithm in [12] is a randomised linear time algorithm for generating MST's and can be used as the MST-generator for the RANSAC procedure. While this approach will be considered in subsequent work, in this paper we shall use an unweighted graph so as to focus on the basic idea of our approach. Thus our method can be summarised as:

[1] Consider a case where an MST has edges between vertices $\{1, 2\}$ and $\{2, 3\}$ but not between $\{1, 3\}$. In such a case we can reach vertex 3 from vertex 1 via vertex 2. Thus the relative motion $\mathbf{M}_{13}$ is given by $\mathbf{M}_{13} = \mathbf{M}_{23}\mathbf{M}_{12}$.

A2 : RANSAC Algorithm for Robust Motion Averaging
Input : $\{\mathbf{M}_{ij1}, \mathbf{M}_{ij2} \cdots, \mathbf{M}_{ijn}\}$ (n relative motions)
Distance threshold $\mathbf{D}_0$ and number of trials $\mathbf{T}$
Output : $\mathbf{M}_g : \{\mathbf{M}_2, \cdots, \mathbf{M}_N\}$ (N image global motion)

```
– Set $G$ : view-graph of relative motions
– Generate MST $\mathbf{e} = MST(G)$
– Solve for global motion $\mathbf{M}_{mst}$ using MST $\mathbf{e}$
– Count number of relative motions within distance $\mathbf{D}_0$ of $\mathbf{M}_{mst}$
– Repeat for $\mathbf{T}$ trials and select MST with maximal count
– Discard relative motions that are outliers for this MST
– Using the inliers solve for $\mathbf{M}_g$ using Algorithm A1
```

4 Examples

To demonstrate the efficacy of robust motion estimation we present an experiment on the well-known MOVI house sequence[2]. This sequence consists of 118 images of a house model and other objects rotated on a turn-table. Fig. 3(a) shows one image from the sequence. As a baseline for comparison, we use the point correspondences of this sequence used in [9].[3] For every possible image pair with more than 20 correspondences we estimated the epipolar geometry using the Eight Point Algorithm of [13]. The camera calibration was estimated using the method outlined in [14] and subsequently the epipolar geometries were decomposed into rotations and translation directions. Instead of applying our approach to the entire set of relative rotations we used a sliding window of 10 images with a shift of 5 images. In other words, we applied the outlier detection algorithm to images 1 to 10, 6 to 15, etc. The RANSAC threshold was set to $0.25°$ and 10000 trials were used. Out of an original set of 2209 relative geometries, only 1130 were selected as inliers. The results of using our method are shown in Fig. 3(b) and (c). In all cases we represent the result as the location of the camera's viewing direction. In Fig. 3(b) the viewing directions of the result of [9] are shown in solid line and the results of an average of all 2209 relative motions is shown as a dashed line. As can be seen there are gross errors in the averaging result due to the presence of outliers. In comparison the motion shown in Fig. 3(c) is the result of our averaging scheme applied to the 1130 inliers detected. Here the correct nature of the sequence is captured implying that outliers were correctly identified and removed, thus demonstrating the effectiveness of our approach to robust motion averaging[4]. In addition to using graph-sampling to identify outliers we can also apply the same MST-based sampling approach on a graph representing all the inliers. This results in a different solution for each MST generated

[2] We are unable to present an analysis of the method's performance on synthetic data due to space constraints.

[3] Thanks to Bogdan Georgescu for providing us with his correspondences and motion estimates for this sequence.

[4] While we do not have any ground truth for this sequence, our results for rotation estimation are on an average within 2 degrees from the estimate of [9]. This is a very good fit given that the estimation of the eight-point algorithm is intrinsically error prone.

and these solutions represent an empirical estimate of the covariance in our estimation process. This is a principled approach in statistics known as *bootstrap*, further details can be found in [6]. In our case we generate 100 such estimates and the covariance of viewing directions were computed. In Fig. 3(c) we show the covariance of the viewing direction for 6 images in the entire sequence. The covariances were exaggerated 25 times to enable easy visualisation. As can be observed, for some images there is larger variance of the viewing direction in a direction orthogonal to the viewing direction of the camera. This implies that for these frames the uncertainty of the rotation estimate is higher in a direction orthogonal to the viewing direction. The ability to estimate the covariance in this manner can be used in further analysis and improvement of the estimates.

5 Conclusions

In this paper we have presented a RANSAC style sampling approach to incorporate robustness into motion averaging algorithms which accurately identifies statistical outliers in a set of relative motions. The effectiveness of the method was demonstrated on a motion estimation problem. Future work will include effective utilisation of confidence information for the relative motions which can be used in the randomised MST approach of [12] and the development of this robust motion averaging approach for image registration and super-resolution, and robotic path planning approaches like *Simultaneous Localisation and Mapping* (SLAM).

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
2. Govindu, V.M.: Combining two-view constraints for motion estimation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2001) 218–225
3. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2004) 684–691
4. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–95
5. Torr, P.H.S., Murray, D.W.: The development and comparison of robust methods for estimating the fundamental matrix. International Journal of Computer Vision **24** (1997) 271–300
6. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall (1993)
7. Varadarajan, V.: Lie Groups, Lie Algebras and Their Representations. Volume 102 of Graduate Texts in Mathematics. Springer-Verlag (1984)
8. Fletcher, P.T., Lu, C., Joshi, S.: Statistics of shape via principal component analysis on lie groups. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2003) 95–101
9. Georgescu, B., Meer, P.: Balanced recovery of 3d structure and camera motion from uncalibrated image sequences. In: European Conference on Computer Vision. (2002) 294–308
10. Levi, N., Werman, M.: The viewing graph. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2003) 599–606

11. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms. MIT Press (2001)
12. Karger, D., Klein, P., Tarjan, R.: A randomized linear-time algorithm for finding minimum spanning trees. Journal of the ACM **42** (1995) 321–328
13. Hartley, R.: In defence of the 8-point algorithm. In: Proceedings of the 5th International Conference on Computer Vision. (1995) 1064–1070
14. Mendonca, P.R.S., Cipolla, R.: A simple technique for self-calibration. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1999) 112–116

Detection of Moving Objects by Independent Component Analysis

Masaki Yamazaki, Gang Xu, and Yen-Wei Chen

Faculty of Information Science and Engineering, Ritsumeikan University, Shiga, Japan
{ymasaki, xu}@cvg.is.ritsumei.ac.jp,
chen@is.ritsumei.ac.jp

Abstract. Detection and tracking of moving objects is very important in various ways. Concerning the detection of moving objects by stationary cameras, the background looks different as the illumination changes. In this paper, we consider a particular image in an image sequence as the sum of a reference image containing the background and a difference image containing the moving objects but not the background. We show that a reference image and difference images can be obtained as the independent components of input images by Independent Component Analysis. Moving objects can then be located on the reference image and the difference images. Experimental results show that the proposed approach produces accurate detection of moving objects even if illumination changes.

1 Introduction

Detection and tracking of moving objects is very important in various ways, such as traffic control and video surveillance. For detecting moving objects, there are mainly three methods. They are background subtraction, frame differencing and optical flow segmentation [1 ~ 3]. Since it takes a lot of computational time to extract optical flow, optical flow based methods does not suit for a real time processing. On the other hand, difference images are generated by a very simple processing, so difference images based methods can be performed in real time. Generally, either a background image obtained in advance or an image taken just previously in the image sequence is used to calculate the difference with the current image. For solving illumination change problems, there are background estimation and histogram adjustment method [11, 12]. But it is very difficult for background estimation to get a background image in a street or behind the crowd. And it occur error detection of moving objects for the object's shadow, spotlight, etc [10, 14].

Recently, some applications of Independent Component Analysis (ICA) that is a statistical data analysis method [4] have been exploited in the field of image processing and computer vision. For example, the face recognition [5, 6], the blind deconvolution of the blurred image [7] and the separating reflections [8, 9].

The input images observed by stationary cameras consist of the background and moving objects, so we consider a particular image in an image sequence as the sum of a reference image containing the background and a difference image containing the moving objects but not the background. We show that a reference image and difference images can be obtained as the independent components of input images by ICA.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 467 – 478, 2006.

Moving objects can then be located on the reference image and the difference images. Experimental results show that the proposed approach produces accurate detection of moving objects even if illumination changes.

Section 2 describes our new formulation of moving object detection problem. Section 3 presents separation by ICA and moving object localization. Section 4 describes experimental results. Section 5 summaries our conclusion.

2 Formulation of Moving Object Detection Problem

For detecting moving objects, we need to separate moving object areas and the background. The easiest way to detect moving objects is to subtract the background image from the input image. In this case, the images ($I_1, I_2, \ldots, I_n$) can be represented by

$$\begin{cases} I_1 = I_{Background} + \vartriangle I_1 \\ I_2 = I_{Background} + \vartriangle I_2 \\ \quad \vdots \\ I_n = I_{Background} + \vartriangle I_n \end{cases} \tag{1}$$

where $I_{Background}$ is the background image and $\vartriangle I_n$ is a difference image between I_n and $I_{Background}$. Note that the *n* input images are separated into one background image and *n* difference images.

Another way to represent the input images is as follows:

$$\begin{cases} I_1 = I_i + \vartriangle I_{1i} \\ I_2 = I_i + \vartriangle I_{2i} \\ \quad \vdots \\ I_i = I_i \\ \quad \vdots \\ I_n = I_i + \vartriangle I_{ni} \end{cases} \tag{2}$$

where I_i acts a reference image that contains the background and $\vartriangle I_{ni}$ is the difference between I_n and I_i that does not contain the background.

An example of two images is shown below in Fig. 1. One of the two images can be understood as the sum of the other image as the reference image and the difference image.

When the lighting condition is constant, the difference images have zero intensity except in the areas of moving objects. Note that in this separation, there are one reference image and *n-1* difference images, and the total number of images remains to be *n*. Also notice that such separation is not unique. Actually, many linear combinations can serve the same purpose.

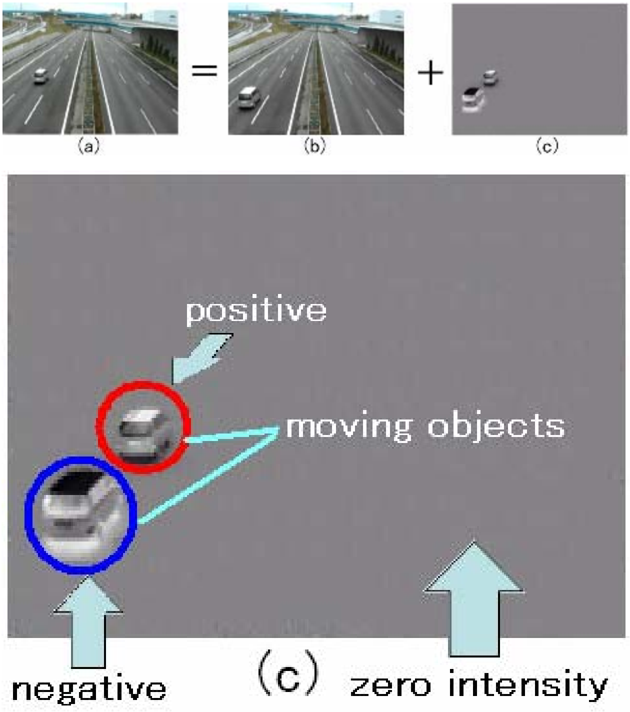


Fig. 1. Representation of a particular image in an image sequence. (a) a particular image in an image sequence. (b) a reference image. (c) a difference image that is subtracted (b) from (a).

Assuming the lighting condition may change, Eq.(2) can be generalized as

$$\begin{cases} I_1 = a_{11} I_i + a_{12} \Delta I_{1i} \\ I_2 = a_{21} I_i + a_{22} \Delta I_{2i} \\ \quad \vdots \\ I_i = a_{i1} I_i \\ \quad \vdots \\ I_n = a_{n1} I_i + a_{n2} \Delta I_{ni} \end{cases} \tag{3}$$

where $(a_{11}, a_{12}, \ldots, a_{n1}, a_{n2})$ are coefficients, which will be computed by ICA.

It is important to observe that moving objects and a background are statistically independent. Therefore, the above separation can be achieved by ICA. In the strict sense, a reference image is not the "pure" background as ICA assumes. In frame image sequence, the reference image is a frame image. That is why the "pure" background cannot be recovered by ICA directly. One merit of using ICA is that the background does not need to be constant. That is the separation will not be affected even if the background changes due to a lighting variation and/or a change in the camera contrast.

3 Separation by ICA and Moving Object Localization

3.1 Brief Introduction of ICA

ICA generalizes the technique of Principal Component Analysis (PCA) [13] and has proven to be a good tool of feature extraction. When some mixtures of probabilistically independent source signals are observed, ICA recovers the original source signals from the observed mixtures without knowing how the sources are mixed. The general model can be described as follows:

We start with the assumption that the observation vectors $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M)^T$ can be represented in terms of a linear superposition of unknown independent vectors $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N)^T$.

$$\mathbf{X} = \mathbf{AS} \tag{4}$$

where **A** is an unknown mixing matrix (M×N) . The goal of ICA is to find a matrix **W**, so that the resulting vectors

$$\mathbf{Y} = \mathbf{WX} \tag{5}$$

recovers the independent vectors **S**, probabilistically permuted and rescaled. **W** is roughly the inverse matrix of **A**.

Before performing ICA, the problem of estimating the matrix **A** can be simplified by a prewhitening of the vectors **X**. The observed vectors **X** is first linearly transformed to other vectors

$$\mathbf{Z} = \mathbf{MX} \tag{6}$$

whose correlation matrix equals unity: $E(\mathbf{Z} \cdot \mathbf{Z}^{\mathbf{T}}) = \mathbf{I}$. This can be accomplished by PCA with

$$\mathbf{M} = \mathbf{D}^{-1/2}\mathbf{V} \tag{7}$$

where the matrix **V** is the eigenvector matrix of the covariance matrix of **X** and the matrix **D** is the eigenvalue matrix of the covariance matrix of **X**. At the same time, the dimensionality of the vectors is reduced. After this transformation we have

$$\mathbf{Z} = \mathbf{MX} = \mathbf{MAS} = \mathbf{BS} \tag{8}$$

where the matrix **B** is the mixing matrix. ICA is performed on the sphered vectors **Z** and the estimated mixing matrix **B** is an orthogonal matrix,

since $E(\mathbf{Z} \cdot \mathbf{Z}^{\mathbf{T}}) = \mathbf{B}E(\mathbf{S} \cdot \mathbf{S}^{\mathbf{T}})\mathbf{B}^{\mathbf{T}} = \mathbf{BB}^{\mathbf{T}} = \mathbf{I}$.

After a prewhitening of the vectors **X**, we can rewrite Eq.(5) to:

$$\mathbf{Y} = \mathbf{WZ} \tag{9}$$

Several ICA algorithms have been proposed for solving **W**. Here we use a neural learning algorithm proposed by Bell & Sejnowski [4]. The algorithm is to maximize the joint entropy by using a stochastic gradient ascent. The gradient update rule for the weight matrix **W** is as follows:

$$\Delta\mathbf{W} = \left(\mathbf{I} + g(\mathbf{Y})\mathbf{Y}^T\right)\mathbf{W} \tag{10}$$

where $g(\mathbf{Y}) = 1 - 2/(1 + e^{-\mathbf{Y}})$ is calculated for each component of **Y**.

3.2 Moving Object Localization and Background Synthesis

From the reference image and the difference images produced by ICA, we can further detect moving objects and synthesize a “pure” background image without moving objects. By comparing each pixel in an original input image with all the difference images, we can determine if the pixel belongs to the background or to a moving object. Since zero intensity pixels in the difference images are background, we simply check each pixel in an original input image that correspond to the other pixel except zero intensity pixel in difference images with thresholding. This way, each input image is divided into the background pixels and the moving object pixels. Excluding the pixels belonging to the moving objects, the original input image has a few empty areas. The colors in these areas can be filled by pasting from other input images where the background is not occluded by moving objects.

4 Experimental Results

In this section, we present the moving object detection results both for a constant background and for a changing background. In these experiments, we used a digital video camera whose frame rate is less than 30 fps with a dimension of 320*240 pixels.

4.1 Constant Background

In the case of two input images, Eq.(4) becomes

$$\mathbf{X} = \mathbf{A}\mathbf{S}$$

$$\begin{bmatrix} \mathbf{x}_1^{\ T} \\ \mathbf{x}_2^{\ T} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^{\ T} \\ \mathbf{s}_2^{\ T} \end{bmatrix} \tag{11}$$

In the example shown in Fig.2, two input images in video image are used and two independent components are separated by ICA.

The estimated weight matrix **A** (Fig.2) is given by

$$\mathbf{A} = \begin{bmatrix} 0.96 & -0.06 \\ 0.99 & 0.94 \end{bmatrix} \tag{12}$$

As can be seen from Fig.2, Fig.2(c) is very close to Fig.2(a), which serves as the reference image. Fig.2(d) is the difference between Figs.2 (b) and (a). This can be verified from Eq.(12). The fact that a_{11} is close to 1 and a_{12} is close to 0 means that Fig.2(a) is almost the same as the reference image. The fact that both a_{21} and a_{22} are close to 1 means that Fig.2(b) is almost the sum of Figs.2 (c) and (d), that is the reference image and the difference image. From these separated images, we can further synthesize a “pure” background image and localize moving objects in each input image, as shown in Figs.2 (g), (e), (f), respectively.

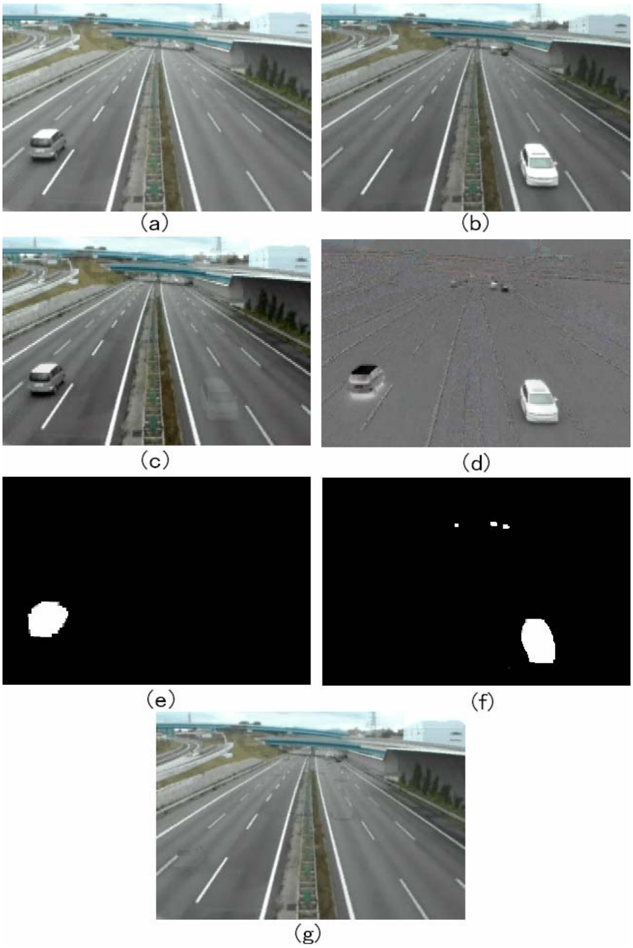


Fig. 2. ICA of two input images. (a), (b) two input images. (c), (d) two separated images by ICA. (e), (f) moving objects. (g) a synthetic background image.

In the case of three input images, Eq.(4) becomes

$$\mathbf{X} = \mathbf{AS}$$

$$\begin{bmatrix} \mathbf{x}_1{}^T \\ \mathbf{x}_2{}^T \\ \mathbf{x}_3{}^T \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1{}^T \\ \mathbf{s}_2{}^T \\ \mathbf{s}_3{}^T \end{bmatrix} \tag{13}$$

In the example shown in Fig.3, three input images in video image are used and three independent components are separated by ICA.

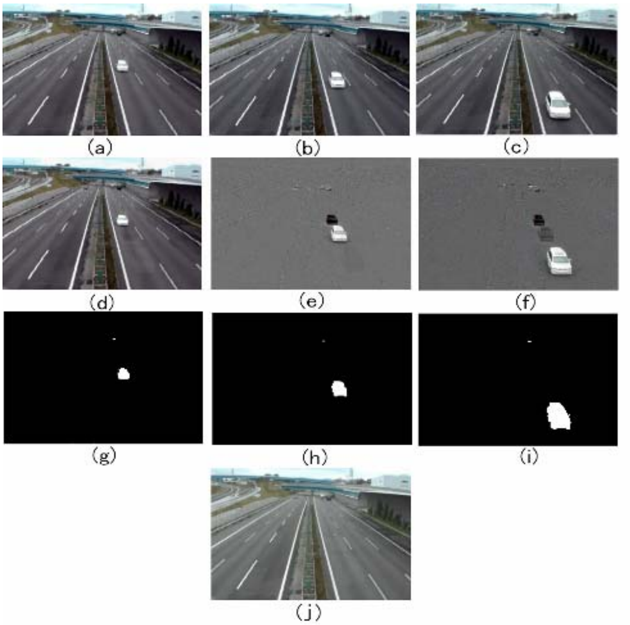


Fig. 3. ICA of three input images. (a), (b), (c) three input images. (d), (e), (f) three separated images by ICA. (g), (h), (i) moving objects. (j) a synthetic background image.

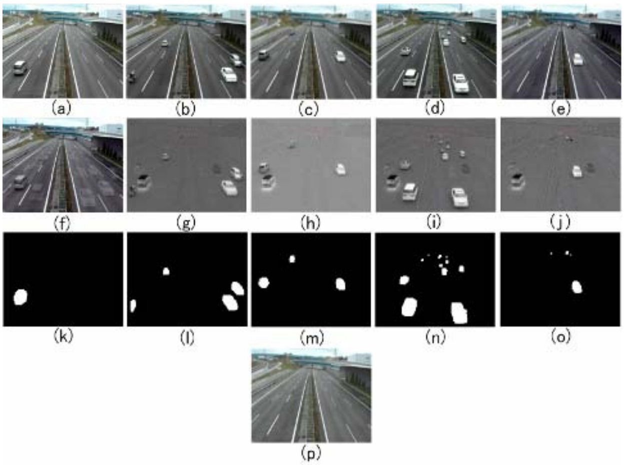


Fig. 4. ICA of five input images. (a), (b), (c), (d), (e) five input images. (f), (g), (h), (i), (j) five separated images by ICA. (k), (l), (m), (n), (o) moving objects. (p) a synthetic background image.

The estimated weight matrix **A**(Fig.3) is given as follows:

$$\mathbf{A} = \begin{bmatrix} 1.04 & -0.03 & -0.06 \\ 0.95 & 1.02 & -0.04 \\ 0.97 & 0.09 & 0.96 \end{bmatrix} \tag{14}$$

As can be seen from Fig.3, Fig.3(d) is very close to Fig.3(a), which serves as the reference image. Figs.3 (e) and (f) are the differences between Figs.3 (a), (b) and (c). This can be verified from Eq.(14). The fact that a_{11} is close to 1, and a_{12} and a_{13} are close to 0 means that Fig.3(a) is almost the same as the reference image. The fact that both a_{21} and a_{22} are close to 1 and a_{23} is close to 0 means that Fig.3(b) is almost the sum of Figs.3 (d) and (e). The fact that a_{31} is close to 1 and that the sum of a_{32} and a_{33} is close to 1 means that Fig.3(c) is the sum of Fig.3(d) and a linear combination of Figs.3 (e) and (f), with Fig.3(f) weighted more.

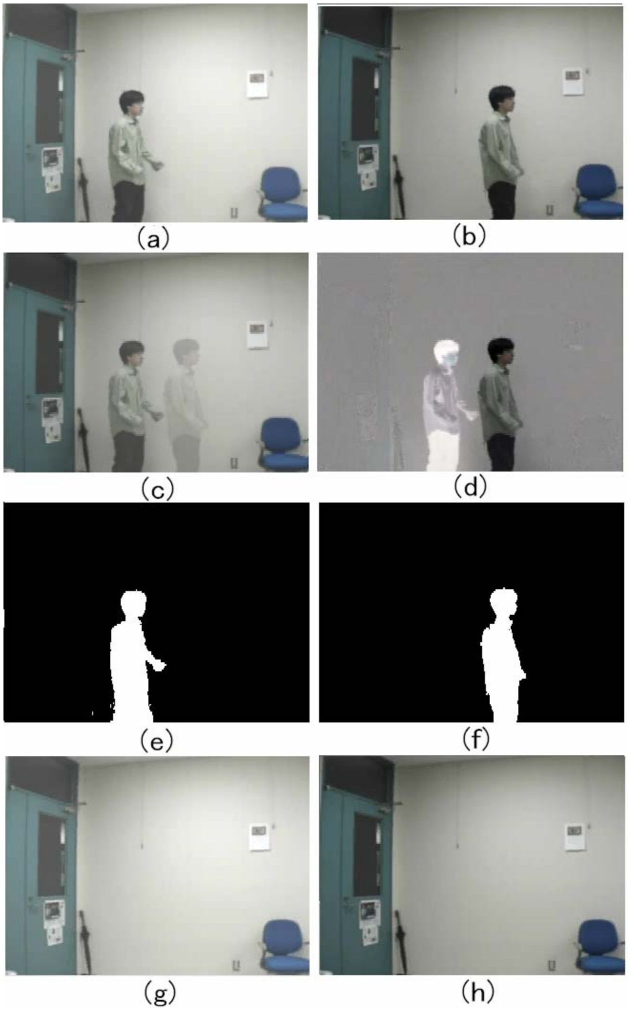


Fig. 5. ICA of two input images. (a), (b) two input images. (c), (d) two separated images by ICA. (e), (f) moving objects. (g), (h) two synthetic background images.

Without giving detailed explanations, we show an example of five input images in video image in Fig.4.

4.2 Changing Background

The method based on ICA is not only effective in the case of constant background but also very effective in the case of changing background due to lighting variations and contrast changes. Two examples are shown in Fig. 5 and Fig. 6. They are the same scene but with different lighting conditions.

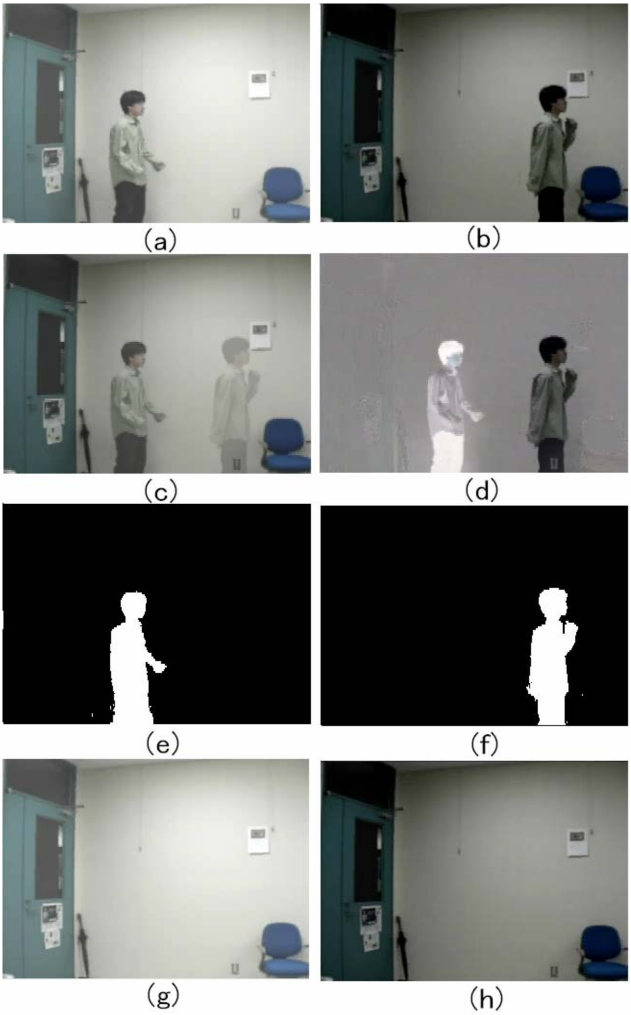


Fig. 6. ICA of two input images. (a), (b) two input images. (c), (d) two separated images by ICA. (e), (f) moving objects. (g), (h) two synthetic background images.

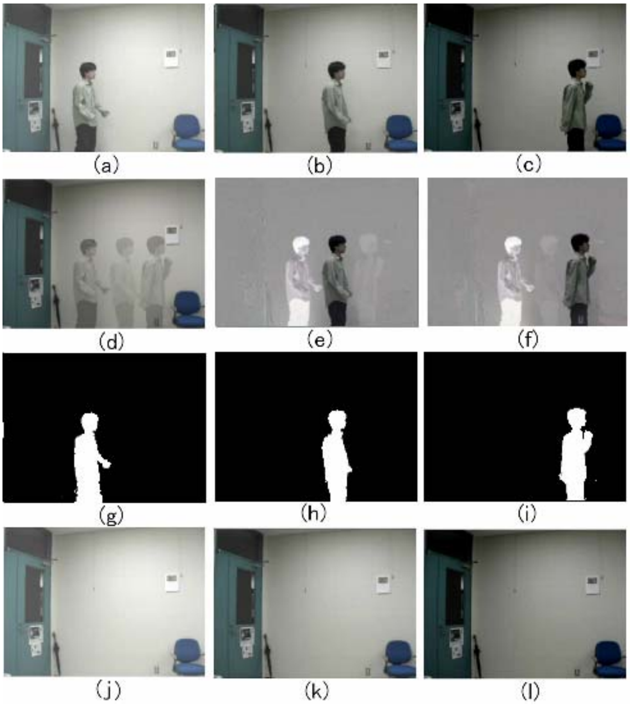


Fig. 7. ICA of three input images. (a), (b), (c) three input images. (d), (e), (f) three separated images by ICA. (g), (h), (i) moving objects. (j), (k), (l) three synthetic background images.

The estimated weight matrix **A**(Fig.5) is given as follows:

$$\mathbf{A} = \begin{bmatrix} 1.12 & -0.09 \\ 0.91 & 0.94 \end{bmatrix} \tag{15}$$

Compared with the example shown in Fig.2, a_{11} and a_{21} are less close to 1, because the background change has to be accounted for by the weights. The more the background changes, the more a_{11} and a_{21} differ from 1, as can be verified by Eq.(16) which is for the example given in Fig.6. From these separated images, we can further synthesize a “pure” background image by adjusting the scale of the pasted empty areas to that of the background and localize moving objects in each input image, as shown in Figs.5,6 (g), (h), (e), (f), respectively.

The estimated weight matrix **A**(Fig.6) is given as follows:

$$\mathbf{A} = \begin{bmatrix} 1.19 & -0.11 \\ 0.82 & 0.96 \end{bmatrix} \tag{16}$$

The same trend can be seen in the case of three input images in video image with a changing background as shown in Fig. 7. The estimated weight matrix **A** (Fig.7) is given as follows:

$$\mathbf{A} = \begin{bmatrix} 1.21 & -0.11 & -0.13 \\ 0.92 & 0.93 & -0.09 \\ 0.81 & 0.06 & 0.91 \end{bmatrix} \cdot \tag{17}$$

4.3 Comparison with Other Related Methods

We compare our proposed method with common frame differencing method in the same scene. As a result of experiments, our proposed method is almost the same performance. With regard to lighting results, it occur error detection of moving objects when partial illumination changes (spotlight etc). These problems, however, can be solved if partial illumination change periods are parameterized. Our proposed method has the merit of dealing with several images at once. That is to say, it is possible to get more information about moving objects.

5 Conclusions

We have presented a new method for detecting moving objects based on ICA. A particular image in an image sequence can be modeled as a linear combination of a reference image containing the background and a difference image containing the moving objects but not the background. The reference image and difference images can be obtained by applying ICA to the input images. Moving objects are then detected by using the separated images by ICA. Experimental results agreed well with the assumed model and have shown that our proposed method is effective in detecting moving objects even if changing lighting conditions except partial illumination changes (spotlight etc).

Many methods for moving object detection have both good points and bad points. So we need design effective algorithm by the problem conditions. Further investigations will extend the proposed algorithm to tracking of moving objects in real time by combining with other algorithm.

References

[1] N. Ohta, "A statistical approach to background substraction for surveillance systems," *Int. Conf. Computer Vision,* vol. 2, pp. 481-486, 1995.

[2] J. Ren, P. Astheimer, and D. Feng, "Real-time Moving Object Detection Under Complex Background," *3rd IEEE Int Symposium on Image and Signal Processing and Analysis*, pp.662-667, 2003.

[3] M. Lucena, J. Fuertes, J. Gomez, N. Perez de la Blanca, and A. Garrido, "Tracking from Optical Flow," *3rd IEEE Int Symposium on Image and Signal Processing and Analysis*, pp. 651-655, 2003.

[4] A. Bell, and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation* 7:1129-1159, 1995.

[5] M. Bartlett, Face Image Analysis by Unsupervised Learning*, Kluwer Academic*, 2001.

[6] M. Bartlett, J. Movellan, and T. Sejnowski, "Face Recognition by independent component analysis," *IEEE Trans. on Neural Networks,* vol. 13, no. 6, pp. 1450-1464, 2002.

[7] S. Umeyama, "Blind Deconvolution of Images Using Gabor Filters and Independent Component Analysis," *Proc. Symp. Independent Component Analysis and Blind Signal Separation*, pp. 319-324, 2003.

[8] H. Farid and E. Adelson, "Separating Reflection and Lighting Using Independent Component Analysis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 262-267, 1999.

[9] S. Umeyama, "Separation of Diffuse and Specular Components of Surface Reflection by Use of Polarization and Statistical Analysis of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 639-647, 2004.

[10] J. Stauder, R. Mech, and J. Ostermann, "Detection of moving cast shadows for object segmentation," *IEEE Transactions on Multimedia*, vol. 1, no. 1, pp. 65-76, 1999.

[11] B. Lee and M. Hedley. "Background Estimation for Video Surveillance," *Image and Vision Computing New Zealand 2002*, pp.315-320, 2002.

[12] A. Elgammal, D.Harwood, and L. Davis, "Non-parametric model for background substraction," *European Conf. Computer Vision,* vol. 2, pp. 751-767, 2000.

[13] F. Torre, and M. Black, "Robust principal component analysis for computer vision," *Int. Conf. Computer Vision,* vol. 1, pp. 362-369, 2001.

[14] T. Haga, K. Sumi, Y. Yagi, "Human Detection in Outdoor Scene Using Spatio-Temporal Motion Analysis," *Int. Conf. on Pattern Recognition*, vol.4, pp.331-334, 2004.

OK-Quantization Theory and Its Relationship to Sampling Theorem

Yuji Tanaka[1], Takayuki Fujiwara[1], Hiroyasu Koshimizu[1], and Taizo Iijima[2]

[1] School of Computer and Cognitive Sciences, Chukyo University,
101 Tokodachi Kaizu-cho, Toyota, Aichi, Japan
yuji@koshi-lab.sccs.chukyo-u.ac.jp
{tfuji, hiroyasu}@sccs.chukyo-u.ac.jp
[2] Professor Emeritus of Tokyo Institute of Technology
tiijima@mb.infoweb.ne.jp

Abstract. OK-Quantization Theory for the digitization in value ensures the reconstructivity of the probabilistic density function of the image. This paper shows some experimental demonstrations to reduce the number of the gray levels, and shows mainly that there is a necessary analytical relationship between sampling and quantization based on the equivalence relationship between two kinds of the integral, *Riemann* and *Lebesgue* integrals for calculating the volume of the image. Experimental demonstrations are also shown in this paper.

1 Introduction

Complete digitization of image $f(x)$ is composed by a pair of digitization processes, sampling and quantization for space(x) and value(f), respectively. Shannon's Sampling Theorem (ST) [1],[2] is a unique mathematical basis for the digitization of the space x, but there is no mathematical basis for the value f. [3],[4],[5] Being inspired by the fact that ST is a reconstruction theorem of the continuous image from the spatially discrete samples of the image, this paper proposes a new mathematical theory for the quantization of the image. This theory, Quantization Theory (QT) is a reconstruction theory for the continuous probability density function(PDF) p(f) of the value f of the image from the digitized PDF. Comparing with the related researches such as the vector quantization, rate distortion theory, Llod-max quantization, Isomichi's inverse quantization, OK-QT is an unique theory from the view point of the mathematical reconstruction basis. This paper investigates briefly an outline of ST for the preparation of the introduction of QT, and QT is proposed as the reconstruction theory of p(f) from the digitized PDF. Executive procedures for estimating p(f) based on a histogram of the given digital image is successively introduced, and some experiments are given to demonstrate the practical impacts of this QT by means of SEM images. Further subjects such as the interference between QT and ST are also pointed as one of the future research subjects and a relationship between OK-QT and ST was presented as an important necessary condition together with some experimental considerations. This was initiated by the fact the volume of the image defined on the *Riemann* integral must be always equivalent to that on the *Lebesgue* integral.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 479 – 488, 2006.

2 The Outline of OK-QT

OK-Quantization theory is a reconstruction basis for the probability density function of the image from its digital data.

2.1 Sampling Theorem and Quantization Theory

ST is a theorem for perfect reconstructing of analog image $f(x)$ from the digital data $f(\Delta x{*}i)$ defined by the given sampling interval Δx. Equation (1) gives the digitized image $f = (f_i)$, where $f_i = f(\Delta x{*}i)$. Here let the Fourier transform of $f(x)$ be $F(u)$.

$$f_i = f(\Delta x * i) \text{ , i= …,-1,0,1,2,3,…} \tag{1}$$

When and only when the Fourier transform $F(u)$ of $f(x)$ is satisfied by eq. (2) (integrable) and $F(u)$ has a cut-off frequency u_c as eq.(3), if and only if the sampling interval Δx be defined by eq. (4), the image can be reconstructed from the digitized image $f = (f_i)$ introduced by eq. (1) as given in eq. (5).

$$\int_{-\infty}^{+\infty} \left| f(x) \right|^2 dx < \infty \tag{2}$$

$$F(u) = 0 \quad u \geq \mathrm{u}_c \tag{3}$$

$$\Delta x \leq 1/2u_c \tag{4}$$

$$f(x) = \sum_{-\infty}^{\infty} f(i * \Delta x) \cdot Sinc\{2\pi(x - i * \Delta x)\} \tag{5}$$

Based on the above discussion, we have proposed a unique que to introduce a new quantization theory, called OK-Quantization as follows. Let the probability density function PDF of the image $f(x)$ be $p(f)$ and its Fourier transform be $P(v)$. If the cut-off frequency of $P(v)$ is v_c as shown in eq. (6), then PDF $p(f)$ can be perfectly reconstructed by eq. (8) from the digitized PDF by the interval Δf defined by eq. (7).

$$P(v) = 0 \quad v \geq v_c \tag{6}$$

$$\Delta f \leq (1/2v_c) \tag{7}$$

$$p(f) = \sum_k p(\Delta f * k) Sinc\{2\pi(f - k * \Delta f)\}. \tag{8}$$

2.2 Estimation of PDF Engineering of OK-Quantization Theory

The only one clue to know the PDF $p(f)$ of an image $f(x)$ is a gray value histogram $h(f)$ of the digital image $f = (f_i)$ preliminarily prepared in advance, i.e. VGA with 8 bit gray scale.

Let us suppose that the gray value histogram $h(f)$ be distributed on finite discrete (= integer) space, and that the minimum interval $\Delta f_{\min}$ of the value f be '1' , and that the maximum value $f_{\max}$ of it be '256'. For example of 8 bit gray scale, $\Delta f_{\min} = 1$ and $f_{\max} = 256$.We introduce a method to estimate $p(f)$ from $h(f)$, which is defined on the finite space $[0, f_{\max}]$, as follows:Let us assume that the PDF $p(f)$ of the given image be a series of *rect*-functions *rect*(*f*) introduced on a finite space [0, $f_{\max}$] as shown in Figure 1, and that the estimated PDF *p*(*f*) can be expressed by eq. (9) on a infinite space.

$$p(f) = \frac{\sum_{n=1}^{f_{\max}} h(\Delta f_{\min} \times n) \times rect(f - n)}{\sum_{n=1}^{f_{\max}} h(\Delta f_{\min} \times n) \times \Delta f_{\min}} \tag{9}$$

Here the definition of *rect* function is shown in eq. (10).

$$rect(x) = \begin{cases} 1 ...... |x| \leq \frac{1}{2} \\ 0 ...... other \end{cases} \tag{10}$$

The PDF *p*(*f*) [*f*:–∞, +∞] has been introduced from the gray value histogram *h*(*f*) [*f*: 0, $f_{\max}$]. Figure 1 shows a *h*(*f*) (dots on the graph) and the estimated PDF *p*(*f*) (trains of *rect* functions). Note that the correspondence between [*f*: 0, $f_{\max}$] and [*f*: –∞, +∞] is one to one mapping only at the integer. As shown in Figure 2, since the Fourier transform *RECT*(*v*) of *rect*(*f*) becomes a *sinc*(*v*), the Fourier transform of PDF *p*(*f*) can be analytically calculated as given in eq. (11).

$$P(\nu) = F[p(f)] = \frac{1}{S} \sum_{n=1}^{f_{\max}} h(\Delta f_{\min} \times n) \exp(-i2\pi n \nu) \frac{\sin(\pi \nu)}{\pi \nu}$$

$$S = \sum_{n=1}^{f_{\max}} h(\Delta f_{\min} \times n) \times \Delta f_{\min} \tag{11}$$

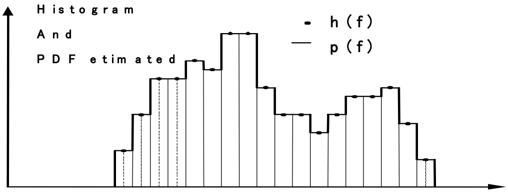


Fig. 1. Gray value histogram and estimated PDF *P*(*f*)

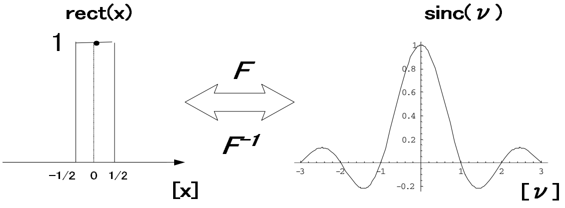


Fig. 2. *rect*[*x*]function and its Fourier transform

2.3 Experimental Consideration

SEM digital image (Semiconductor Resist, 256 gray value levels) shown in Figure 3 is used. The gray value histogram $h(f)$ of this image is shown in Figure 4.

Figure 5 shows the Fourier transform $P(v)$ of PDF $p(f)$ of the image $f(x)$ given in Figure 3. By means of the procedure preliminarily given in [6], the cut-off frequency was extracted at v_c = 0.055 [line pairs / 1 gray scale unit], and therefore the best quantization pitch Δf was decided by OK-Quantization Theorem as follows:

$$\Delta f = 1/(2v_c) = 1/(0.11) = 9.09 \fallingdotseq 9 \text{ [gray scale unit]} \tag{12}$$

It was known from this result that the quantization pitch Δf =9 is (necessary and) sufficient for reconstructing PDF $p(f)$ estimated in Figure 4, and that the number of gray levels (necessary and) sufficient for the image given in Figure 3 could be reduced to 28 because 256 / 9≒28.

2.4 Major Three of the Expected Subjects in OK-QT Are as Follows

(1) Restoration of PDF and Visual Evaluation

OK-QT does not guarantee the visual evaluation of a image directly. But as known from SEM image with 32 gray levels shown in Figures 6(a), degradation is hardly seen. Therefore , a practical applicability of OK-Quantization Theory to the quantization methodology was clearly suggested. For example, as known in SEM image with 8 gray levels, a false outline occurs and degradation is clear like Figures 6(b).

(2) OK-QT and Reverse Quantization

OK-Quantization theory will be effective in the reverse quantization procedure from the following aspect: When the image with full number of gray levels is restored from the image with the reduced number of gray levels, the gray histogram which is completely restored from the reduced histogram by OK-Quantization theory will regulate the gray level restoration process.

(3) Interference Problem Between ST and OK-QT

In general, if the digitization in value is applied to a image $f(x)$ based on OK-QT, the size of the digitized image will become larger than that of the digitized image where the digitization in space is primary applied. Therefore, the digitization process S(Q(f)) to a image $f(x)$ provides the different digital image of the reverse process given by Q(S(f)) This means that ST and OK-QT must be discussed simultaneously. This problem is one of the most important coming subjects.[8]

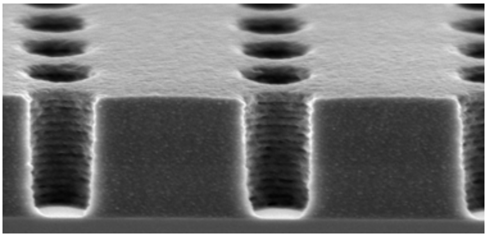


Fig. 3. SEM Image (256gray levels)

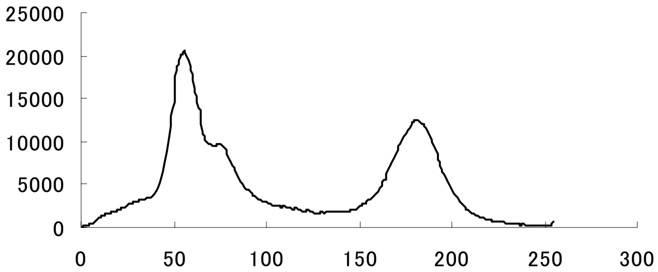


Fig. 4. Gray value histogram h(*f*)

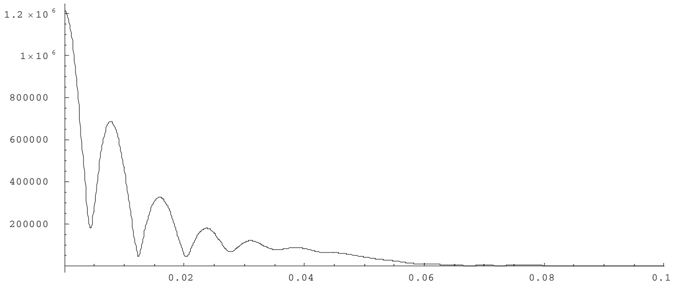


Fig. 5. Fourier transform *P*(*v*)of SEM image of *p*(*f*)

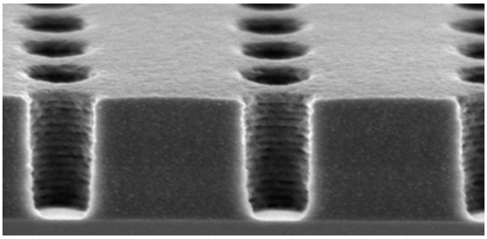


Fig. 6(a). SEM Image (32 gray levels)

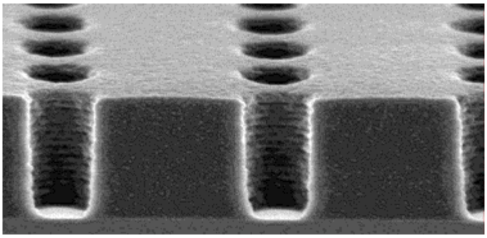


Fig. 6(b). SEM Image (8 gray levels)

3 A Necessary Condition Between ST and OK-QT

3.1 Equivalence Between *Riemann* and *Lebesgue* Integrals of the Image Volume

We discuss here how OK-QT[6],[10][11][12] closely relates to ST. Let us imagine to calculate the volume of the given image from the image $f(x)$ and the probability density function $p(f)$ of its gray value. The volume defined on the image $f(x)$ is provided by *Riemann* integral, and that on the PDF $p(f)$ is provided by *Lebesgue* integral. Then we can extract an analytical relationship between them. Hereafter a new notation ω, $\omega=2\pi u$, is used for convenience.

3.2 Sampling Theorem and Riemann Integral

In this section, let us denote a image as $f(x)$ and a image must be satisfied by eq.(13).

$$f(x) \geq 0 \qquad \int_{-\infty}^{\infty} f(x)dx < \infty \tag{13}$$

Here let the Fourier transform $F(\omega)$ of a image $f(x)$ be band-limited at the cut-off frequency W by eq.(14).

$$\begin{aligned} F(\omega) &= \int_{-\infty}^{\infty} f(x)e^{-i\omega x}dx \qquad (0 \leq |\omega| < W) \\ &= 0 \qquad (W < |\omega| < \infty) \end{aligned} \tag{14}$$

The digital image $f(x_n)(n = 0,\ \pm 1,\ \pm 2,\ \cdots)$ is defined at every sample points as eq.(15).

$$x_n = \frac{n\pi}{W} \quad (n = 0,\ \pm 1,\ \pm 2,\ \ldots) \tag{15}$$

The analog image $f(x)$ can be reconstructed from the digital image by eq.(16) based on Sampling Theorem.

$$f(x) = \sum_{n=-\infty}^{\infty} f(x_n) \frac{\sin(x - x_n)W}{(x - x_n)W} \tag{16}$$

This is the contents of Sampling Theorem known well.

Now, if the integration formula of *Fourier* is applied, the formula (17) will be proved easily.

$$\int_{-\infty}^{\infty} \frac{\sin Wx}{Wx} dx = \frac{\pi}{W} \tag{17}$$

As easily known, the integral of the *sinc* function becomes π/W, the volume I_R of the image can be derivated as eq.(18) based on the *Riemann* integral scheme, and as a result, I_R can be expressed by the digital image.

$$I_R \equiv \int_{-\infty}^{\infty} f(x)dx = \frac{\pi}{W}\sum_{n=-\infty}^{\infty} f(x_n) \tag{18}$$

It becomes clear that the value of definite integral I_R is expressed with the sample value $f(x_n)$ of a function and the cut-off frequency W.

3.3 Overview of OK-Quantization Theory

Let a measure D be defined by eq.(19) such that the value of a image $f(x)$ comes between f_1 and f_2 in the image domain x.

$$D\left[x \middle| f_1 \le f(x) < f_2\right] \tag{19}$$

Then a function $h(f)$ of the value of the image can be expressed by eq.(20).

$$h(f) \equiv \lim_{\delta f=0} D\left[x \middle| f \le f(x) < f + \delta f\right] \Big/ \delta f \tag{20}$$

Since *Fourier* transform $H(v)$ of $h(f)$ is band-limited, $H(v)$ is represented as eq.(21).

$$\begin{aligned} H(v) &= \int_{-\infty}^{\infty} h(f) e^{-ivf} df \quad (0 \le |v| \le V) \\ &= 0 \qquad (V < |v| < \infty) \end{aligned} \tag{21}$$

Just in the same way of ST, the digitized data $h(f_r)$ of $h(f)$ $(r = 0, \pm 1, \pm 2, ...)$ sampled at every f_r introduced by eq.(22) will reconstruct $h(f)$ as eq.(23).

$$f_r = \frac{r\pi}{V} \quad (r = 0, \pm 1, \pm 2, \dots) \tag{22}$$

$$h(f) = \sum_{r=-\infty}^{\infty} h(f_r) \frac{\sin(f - f_r)V}{(f - f_r)V} \tag{23}$$

It means that this result had achieved Quantization of the original function $f(x)$. This is the outline of Quantization Theory.

3.4 OK-Quantization and *Lebesgue* integral

Let us imagine to calculate the volume of the image by way of the integral of $h(f)$. It is promising to know that this integral is equivalent to the *Lebesgue* integral of the image $f(x)$ as follows: If a measure $S(f)$ is introduced as eq.(24) by using the measure D in eq.(19), the volume I_L of the image can be formulated as eq.(25).

$$S(f) = D\left[x \middle| f \le f(x) < \infty\right] = \int_f^{\infty} h(f')df' \tag{24}$$

$$I_L \equiv \int_0^{\infty} S(f)df \tag{25}$$

Substituting S(f) by eq.(24) and executing partial integral, the volume I_L can be represented simply by $h(f)$ as shown in eq.(26).

$$\begin{aligned} I_L &= \int_0^{\infty} df \int_f^{\infty} h(f')df' = \int_0^{\infty} f\ h(f)df \\ &= \int_{-\infty}^{\infty} f\ h(f)df \end{aligned} \tag{26}$$

As $h(f)$ can be replaced by the digitized one by eq.(23), I_L can be represented by eq.(27) and be derivated as eq.(28).

$$\int_{-\infty}^{\infty} f \ \frac{\sin(f - f_r)V}{(f - f_r)V} df \qquad (g = f - f_r)$$

$$= \frac{1}{V} \lim_{L \to \infty} \int_{-L}^{L} \sin \ gV \ dg + f_r \int_{-\infty}^{\infty} \frac{\sin \ gV}{gV} dg \tag{27}$$

$$= \frac{\pi}{V} f_r = \left(\frac{\pi}{V}\right)^2 r \tag{28}$$

Finally the volume of the image can be represented by *h*(*f*$_r$) and cut-off frequency *V*. as shown in eq.(29).

$$I_L = \sum_{r=-\infty}^{\infty} h(f_r) \int_{-\infty}^{\infty} f \ \frac{\sin(f - f_r)V}{(f - f_r)V} df$$

$$= \left(\frac{\pi}{V}\right)^2 \sum_{r=-\infty}^{\infty} r \ h(f_r) \tag{29}$$

It is clear that the value of I_L becomes settled from this formula only with the density function *h*(*f*$_r$) of frequency and the cut-off frequency *V* about a function value.

4 Relationship Between ST and OK-Quantization

The volume of the image I_R based on *Riemann* integral is originally equivalent to I_L based on *Lebesgue* integral as shown in eq.(30), therefore we can have an interesting relationship given in eq.(31). For a given image, the cut-off frequency *W* in image space must be analytically constrained by the cut-off frequency *V* in PDF space.

$$I_R \equiv I_L \tag{30}$$

$$\frac{\pi}{W} \sum_{n=-\infty}^{\infty} f(x_n) = \left(\frac{\pi}{V}\right)^2 \sum_{r=-\infty}^{\infty} r \ h(f_r) \tag{31}$$

Also it is easily known that this equation could be extended as eq.(32) in 2-dimensional image. *W* and *W'* are the cu-off frequency for *x* and *y* axis, respectively.

$$\frac{\pi^2}{WW'} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(x_m, y_n) = \left(\frac{\pi}{V}\right)^2 \sum_{r=-\infty}^{\infty} r \ h(f_r) \cdot \tag{32}$$

5 Experiments and Considerations

Modifying eq.(32), we get eq.(33) to investigate experimentally how 1/*WW'* is constrained by the change of *V*.

$$\frac{\pi^2}{WW'} = \left(\frac{\pi}{V}\right)^2 \sum_{r=-\infty}^{\infty} r \ h(f_r) \Big/ \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(x_m, y_n) \tag{33}$$

Figures 7 and 8 are a set of examples, where the upper is the image and the lower is the respective graph in which the horizontal axis represents the value *V* and the vertical one is the value of the right term of eq.(32). It is generally known that *W* and *V* are approximately linear and that the shape of the graph is strictly dependent to the respective property of the image. This means theoretically that the finer resolution in space requires the finer resolution in value.

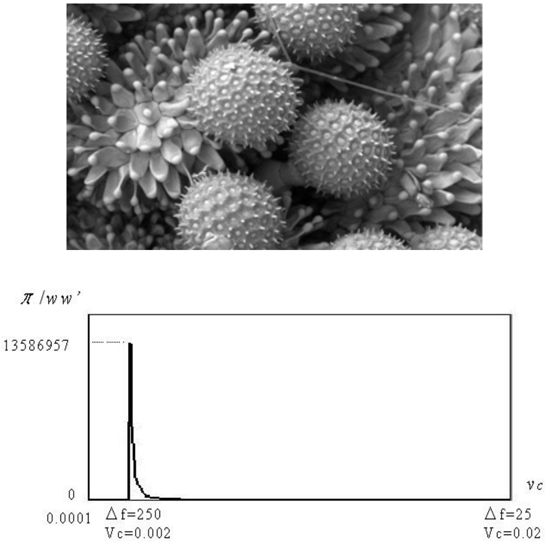


Fig. 7. Input image (flower pollen) and experimental result

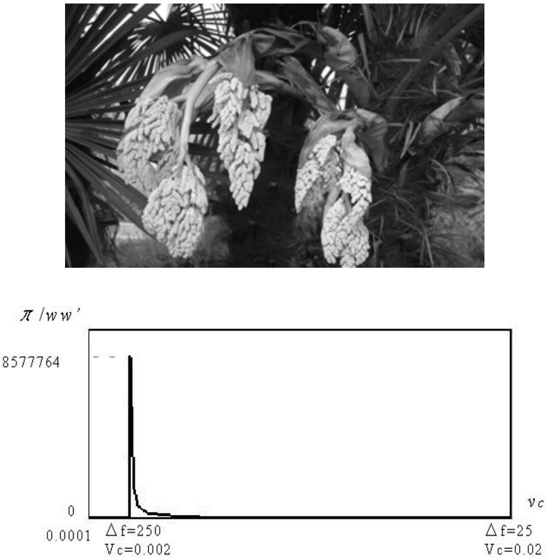


Fig. 8. Input image (plant) and Experimental result

6 Conclusion

In this paper, after considering the outline of OK Quantization Theory briefly, a theoretical relationship between OK-QT and Sampling Theorem was discussed mathematically. From the view point of the equivalence between *Riemann* and *Lebesgue* integrals of the volume of the image, we introduced a necessary condition between cut-off frequencies W and V in the respective spatial and gray value domains. We also showed an experimental demonstration of this necessary condition. In the future, it is expected to introduce a practical procedure for the simultaneous design of sampling and quantization.

References

1. C.E. Shannon, et al.: The Mathematical Theory of Communication, Univ. Illinois Press (1949)
2. Jun Hasegawa, et.al.: The Mathematical Theory of Communication, Meiji Tosho (1969)
3. Kazuo Nakata: Vector Quantization of Vowel and Image Signals, Journal of Measurement and Control, Vol.25, No.6, pp.517-52 (Jun.1986)
4. Yoshinori Isomichi: Exercise in Information Theory, pp.53-55 (Corona Publishing Company) (1983)
5. Hiroyasu Koshimizu, et.al.: A Practical Method for Estimating Aliasing Error in Image Processing, Trans. IEICE, Vol.61-D, No.6, pp.443-444 (Jun.1978)
6. Hiroyasu Koshimizu, et.al: Proposal of Quantization Theorem, Proc.VIEW2002, 1-1 (Dec.2002) (Yokohama)
7. Osamu Oteru: Basic Electric Measurement, pp.236 - 237, (Ohm Publishing Company) (1966)
8. Taizo Iijima(private letter): Considerations on OK-Quantization Therem by Examples, (Feb.18, 2003)
9. Hiroyasu Koshimizu: On A Mathematical Theory of Quantization — How should image gray value be digitized ? –, FCV2003, Invited paper (Feb.6,2003) (Jeju, Korea)
10. Hiroyasu Koshimizu: On Proposal Of Quantization Theorem and Its Experimental Consideration- Theory of image gray scale dispersion -, IEICE(PRUM2003-66)(May.2003)
11. Y.Tanaka,T.Fujiwara,H.Koshimizu, and T.Iijima : ”A Relationship between OK-Quantization and Sampling Theorem and Its Experimental Consideration”, QCAV2005, PP399-404, (May.2005)
12. Yuji Tanaka, Takayuki Fujiwara ,Hiroyasu Koshimizu, Munetoshi Numada: ”OK-Quantization Method and Its Theoretical and Experimental Properties”, MIRU2005, PP1495-1502, (July.2005)

Contour Matching Based on Belief Propagation

Shiming Xiang, Feiping Nie, and Changshui Zhang

State Key Laboratory of Intelligent Technology and Systems,
Department of Automation, Tsinghua University, Beijing 100084, China
{xsm, nfp03, zcs}@mail.tsinghau.edu.cn

Abstract. In this paper, we try to use graphical model based probabilistic inference methods to solve the problem of contour matching, which is a fundamental problem in computer vision. Specifically, belief propagation is used to develop the contour matching framework. First, an undirected loopy graph is constructed by treating each point of source contour as a graphical node. Then, the distances between the source contour points and the target contour points are used as the observation data, and supplied to this graphical model. During message transmission, we explicitly penalize two kinds of incorrect correspondences: many-to-one correspondence and cross correspondence. A final geometrical mapping is obtained by minimizing the energy function and maximizing a posterior for each node. Comparable experimental results show that better correspondences can be achieved.

1 Introduction

1.1 Background and Related Work

Shape matching is an essential and critical topic in computer vision. The task of shape matching is to find the point-to-point geometric correspondences between two sets of 2D or 3D points [1] to establish an optimal aligning transformation for shape-based image analysis or object recognition [2, 3]. Many approaches to shape matching have been proposed [1]. However, matching shapes with local non-rigid deformations and different topologies is still a tough issue.

In real applications, the shapes of the objects to be matched can be specified by a set of structured or unstructured points [4, 5, 6, 7, 8, 10]. This paper considers the special case where the shape is represented by a single closed contour.

Various efforts on improving contour matching are concentrated on contour feature and matching algorithm. To obtain accurate correspondences, one needs shape descriptors with local rich descriptive power [3, 9, 11].

Dynamic programming (DP) is commonly used to match contours [7, 12, 13]. By means of DP, one can get a mapping with order preserving. But many-to-one correspondences may appear (Figure 1(a)). So a lot of continuous points on the target contour may miss to be matched. Frenkel et al. introduce a continuous formulation for this problem by using Sethian's fast matching method [12].

Another popular optimization strategy is energy or distance minimization [4, 5, 14]. Chui et al. choose to relax the constraints on the correspondence to construct a soft mapping [4] for minimizing the thin-plate-spline (TPS) energy.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 489–498, 2006.

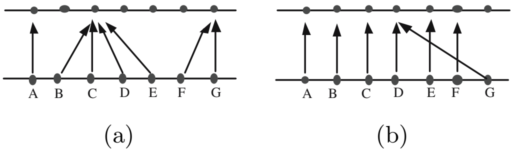


Fig. 1. Incorrect correspondences: (a) many-to-one correspondence and (b) cross correspondences

Coughlan et al. formulate the template shape contour as a Bayesian graphical model [6]. Belief propagation (BP) is performed on this model to match the template to the image [5]. These approaches are very suitable for the sets of unstructured or orderless points. But directly applying them to contour matching may require one to deal with the case of cross correspondence (Figure 1(b)), that is, neighboring points are no longer neighbors after mapped. Grauman et al. use the low-distortion embedding of the earth mover's distance as the minimum cost for contour matching [14]. However, there also lacks of an explicit mechanism to punish cross correspondences.

1.2 Overview

Contour can be defined in the form of a closed sequence of points. Thus order relationship is a natural characteristic for this point sequence. By treating each point as a node, we can get an undirected loopy graph. Each node on the loop has two neighbors. It should share its two neighbors' matching information. The node 'A' and node 'C', for example in Figure 1, should simultaneously transfer information to node 'B'. therefore, it is necessary to introduce a mechanism of dual way message transmission. This treatment is essentially different from DP based frameworks, where the information is transferred in a single way from 'A' to 'B', then to 'C' in the first phase of cost aggregation, and inversely collected from 'C' to 'B', then to 'A' in the second phase of back-tracing.

Since neighboring points on the source contour should be mapped to the neighboring points on the target contour in an order preserving way, the mapped indexes provide the information for introducing penalty term to local messages.

Motivated to the above analysis, in this paper we formulate contour matching as a probabilistic inference problem. The inference is implemented on a loopy graph model by using Bayesian belief propagation algorithm [15].

To obtain an accurate point-to-point mapping, local shape descriptor with rich descriptive power is desired for calculating the distance between a pair of points to be matched. In this paper, we use shape context [3] and curvature information to extract the features of the points.

2 Local Shape Descriptor and Distance Measurement

The shape context [3] has been shown to be a powerful tool for representing shapes. For a single contour point, called as reference point, its shape context is a log-polar histogram of the relative coordinates of the remaining points. The shape context summarizes global shape in a rich and local descriptor. Since each point can be associated with a histogram, we can get a detailed description about the shape perception.

Invariance to translation is intrinsic to the shape context. To achieve scale invariance, all radial distances are normalized by the median distance between all the point pairs [3].

The drawback of shape context descriptor is that the log-polar coordinate system makes it more sensitive to the positions near the reference point [11]. In fact, it is unable to robustly reflect the local geometrical property at the reference point very well. But the degree of curvature is only related to a few neighbor points and can be easily calculated. We use it as an additional feature.

Let point P_S^i belong to contour S, and P_T^j belong to T, the distance between P_S^i and P_T^j is computed as:

$$d(P_S^i, P_T^j) = \chi^2(\mathbf{C}_S^i, \mathbf{C}_T^j) + s_1 d_{s2}(\mathbf{C}_S^i, \mathbf{C}_T^j) + s_2 d_k(\kappa_S^i, \kappa_T^j) + s_3 d_{k2}(\kappa_S^i, \kappa_T^j) \quad (1)$$

where $\mathbf{C}_S^i$ and κ_S^i denote the shape context and the curvature of point P_S^i, respectively. $\mathbf{C}_T^j$ and κ_T^j have the same meanings as $\mathbf{C}_S^i$ and κ_S^i. s_1, s_2 and s_3 are weighting parameters, which are all manually set as 0.001.

In (1), $\chi^2(\mathbf{C}_S^i, \mathbf{C}_T^j)$ and $d_{s2}(\mathbf{C}_S^i, \mathbf{C}_T^j)$ are calculated as the χ^2 test statistics and the two order derivative of the shape context cost at the pair point of (P_S^i, P_T^j) [11, 16], $d_k(\kappa_S^i, \kappa_T^j)$ and $d_{k2}(\kappa_S^i, \kappa_T^j)$ are the curvature cost and the two order derivative of the curvature cost. The reason here we use the two order derivatives is that neighboring points on S should also be neighboring after matched to T.

3 Contour Matching with Belief Propagation

The problem we consider can be described as follows. Let C_s and C_t be a source contour and a target contour. Suppose C_s is defined as a set of m points, $\{P_s^0, P_s^1, \cdots, P_s^{m-1}\}$, and C_t is defined as a set of n points, $\{P_t^0, P_t^1, \cdots, P_t^{n-1}\}$. For simplicity, we use two indicator sets to denote them, $P \overset{\Delta}{=} \{0, 1, \cdots, m-1\}$ and $L \overset{\Delta}{=} \{0, 1, \cdots, n-1\}$. The task now is to assign a label $f_p \in L$ to each point $p \in P$.

3.1 The Max-Product Algorithm

For each point in P, we first assign a node to it and then orderly connect all the nodes to construct an undirected loopy graph. In the setting of probabilistic inference, each node is called a hidden node, which is associated with a state variable. An edge of this undirected graphical model describes the compatibility relationship between the two hidden nodes [15]. By attaching an additional node to each hidden node to transfer observation information, a graph model is constructed as illustrated in Figure 2.

A clique of this model contains a node and its two neighbors. The description and analysis on this model with pairwise cliques of belief propagation become

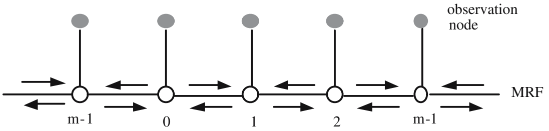


Fig. 2. A loopy graph with hidden nodes and observation nodes for contour matching

more specific and simpler [15]. In an iteration, each node sends a message to each of its two neighbors and receives a message from each neighbor.

Let f_i and f_j be two state variables of two neighboring nodes i and j in the graphical model. Suppose y_i be the associated observation node of hidden node i. We denote by m_{ij}^t the message that node i sends to j at time t, by $m_{ii}(f_i)$ the message that y_i sends to i, and by $b_i(f_i)$ the belief at i. The max-product update rules [15], which control the quality of a labelling, are as follows:

$$m_{ij}^t(f_j) \leftarrow \alpha \max_{f_i}(V_{ij}(f_i, f_j) \cdot m_{ii}(f_i) \cdot \prod_{k \in N} m_{ki}^{t-1}(f_i)) \tag{2}$$

$$b_i(f_i) \leftarrow \alpha \cdot m_{ii}(f_i) \cdot \prod_{k \in N(i)} m_{ki}^t(f_i) \tag{3}$$

where α denotes a normalizing constant, $N = N(i)\backslash j$, denotes the neighbors of i other than j and $V_{ij}(f_i, f_j)$ is a potential function of assignments f_i and f_j, which is the cost of assigning labels f_i and f_j to two neighboring nodes. Actually, $V_{ij}(f_i, f_j)$ is the penalty term to penalize the cross correspondence.

Following the idea used in [17], the equivalent computation can be implemented with negative log probabilities. We rewrite (2) and (3) as follows:

$$m_{ij}^t(f_j) = \min_{f_i}(V_{ij}(f_i, f_j) + m_{ii}(f_i) + \sum_{k \in N} m_{ki}^{t-1}(f_i)) \tag{4}$$

$$b_i(f_i) = m_{ii}(f_i) + \sum_{k \in N(i)} m_{ki}^t(f_i) \tag{5}$$

When using negative log probabilities, all the message vectors are initialized to zero. During message transmission, observation nodes do not receive messages and they always transmit the same vector. After iterations, the label f_i^* that minimizes $b_i(f_i)$ is finally selected as the optimal assignment of the i^{th} node.

The principal of the assignment is intrinsically equivalent to Pearl's rule of finding maximum a posterior [18]. The convergence has not been proven, but several groups have recently reported excellent experimental results by running the max-product algorithm on graphs with loops (for details through [15, 18]).

3.2 Computing Messages

Note that the hidden node i is associated with the point of P_S^i, and its assignment f_i is connected with the point of $P_T^{f_i}$. Therefore, in negative log probability framework, $m_{ii}(f_i)$ can be calculated as the distance between points P_S^i and $P_T^{f_i}$. According to (1), we have:

$$m_{ii}(f_i) = d(P_S^i, P_T^{f_i}) \tag{6}$$

The potential function $V_{ij}(f_i, f_j)$ is related to the cost of the assignment discontinuity in the MRFs for early vision. However, for contour matching, if two neighboring points are assigned to a same label and the cost is zero [17], the situations, that a few continuous points are mapped together to a same point (Figure 1(a)), would appear with higher possibility. Actually, the continuity here is referred to as the order of the assignment. For two neighbor nodes i and j, we hope $\|f_i - f_j\| = 1$ holds when $\|i - j\| = 1$. When a cross correspondence appears ($\|f_i - f_j\| > 1$), a bigger penalty should be given. Thus we define the following cost function:

$$V_{ij}(f_i, f_j) = \begin{cases} 0 & \text{if } \|f_i - f_j\| \ = 1 \\ s & \text{if } f_i = f_j \\ s \cdot \ \|f_i - f_j\| & \text{otherwise} \end{cases} \tag{7}$$

where s is the increasing coefficient and $\|f_i - f_j\|$ denotes the distance of two labels, which is measured on a loop with n labels.

$$\|f_i - f_j\| = \begin{cases} \|f_i - f_j\| & \text{if } \|f_i - f_j\| < n/2 \\ n/2 - \|f_i - f_j\| & \text{otherwise} \end{cases} \tag{8}$$

Note that (7) holds only on the assumption that m is equal to n . When this condition does not meet, we can insert dummy points to one of the point sets such that $m = n$ satisfies. To this end, we copy the points according to a uniform step. For example, when we want to expand a set of six points to a set of eight points, we need to copy two points. The result is: 0, 1, 2, 2, 3, 4, 5, 5.

3.3 Updating Messages

Our model is a bipartite graph. Therefore, the belief propagation can be alternatively performed on two node subsets [17]. Let $P = A \cup B$ ($A \cap B = \emptyset$). Now the message is updated as follows [17]:

$$m_{ij}^t(f_j) = \begin{cases} m_{ij}^t(f_j) & \text{if } i \in A \text{ (if } i \in B) \\ m_{ij}^{t-1}(f_j) & \text{otherwise} \end{cases} \tag{9}$$

It is necessary for negative log probability framework to normalize the message vectors when updating. According to (2) and (4), the normalization for each message vector can be implemented by translating all the elements such that their mean is zero.

4 Fast Algorithm

In each iteration, the performance of the max-product algorithm executes in $O(mn^2)$ time. Felzenszwalb's work [17] shows that the computation time can be reduced from $O(mn^2)$ to $O(mn)$ and one can also get good results. This Section develops a fast algorithm for our contour matching framework introduced in Section 3.

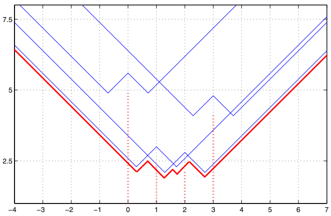


Fig. 3. The computation of the lower envelope of 4 'W's for developing fast algorithm

As can be seen, only the first term in (4) is related to f_j. Thus we have:

$$m_{ij}^t(f_j) = \min_{f_i} \left(V_{ij}(f_i, f_j) + h(f_i)\right) \tag{10}$$

where $h(f_i) = m_{ii}(f_i) + \sum\limits_{k \in N(i) \backslash j} m_{ki}^{t-1}(f_i)$. Given a f_i, the curve of the function $(V_{ij}(f_i, f_j) + h(f_i))$ has a similar form of character 'W' (see Figure 3). It is the lower envelope of two 'V' functions: $V_1 = (\tilde{V}_{ij}(f_i - 1, f_j) + h(f_i))$ and $V_2 = (\tilde{V}_{ij}(f_i + 1, f_j) + h(f_i))$, here $\tilde{V}_{ij}(f_i, f_j) = s \cdot \|f_i - f_j\|$.

Thus we can first calculate the minimum of 'V_1' functions located at '-1' and then calculate that of the 'V_2' functions located at '+1'. As a result, we get two mappings. The best one can be finally selected from two results. Now we give the steps of our fast algorithm:

Step 1: Let $m_1(i) = h(i+1), i = 0, \cdots, n-2$, $m_1(-1) = h(0)$, $m_1(n-1) = h(n-1) + s$, $m_1(n) = h(n-1) + 2 \cdot s$;
Step 2: Let $m_2(i) = h(i-1), i = 1, \cdots, n-1$, $m_2(-1) = h(0) + 2 \cdot s$, $m_2(0) = h(0) + s$, $m_2(n) = h(n-1)$;
Step 3: Do 1D distance transform on $m_1$and m_2 [17];
Step 4: Let $m_{ij}^t(f_j, V_1) = \min(m_1(f_j), \min\limits_{f_i} h(f_i))$ and $m_{ij}^t(f_j, V_2) = \min(m_2(f_j), \min\limits_{f_i} h(f_i))$, $f_j = 0, 1, \cdots, n-1$.
Step 5: Based on $m_{ij}^t(f_j, V_1)$ and $m_{ij}^t(f_j, V_2)$, run twice max-product algorithm and select the best of the two mappings.

The above algorithm can be further improved since the arrays m_1 and m_2 are similar to each other, except the first two and the last two elements. Thus, we need to perform the distance transformation only once. The steps can be rewrited as follows:

Step 1: Do 1D distance transform on h;
Step 2: Let $m(i) = \min(h(i-1), h(i+1)), i = 1, 2, \cdots, n-2$, $m(0) = \min(h(0) + s, h(1))$ and $m(n-1) = \min(h(n-2) + s, h(n-1) + s)$;
Step 3: Let $m_{ij}^t(f_j) = \min(m(f_j), \min\limits_{f_i} h(f_i))$;
Step 4: Run max-product algorithm.

However, the above operation can not be applied to the labels located on a circle. To this end, we extend h from two sides respectively to get a new one with $2n$ elements. Actually, the real labels of the new h array are:

$$([n/2], \cdots, n-1, 0, 1, \cdots, n-1, 0, 1, \cdots, [n/2]-1)$$

In an iteration, we first perform the above fast algorithm on new h. Then for each label we select the smaller one from the two beliefs. Thus the length of h is reduced to the original level.

5 Experimental Evaluation

In this Section we describe the experimental evaluation of our contour matching based on belief propagation and compare it to two matching algorithms: the standard DP and the soft correspondence approach (SC) used in [4]. The reason that we do not use the whole TPS-based shape matching framework to do comparison is that the performance in [4] is based on an interim point-set which is warped from the source shape by the estimated transform parameters. Thus for TPS-based shape matching framework, a good initial matching is very important. Different from the work in [4], when iteratively performing SC, we use the local descriptors in Section 2 to replace the simple distance description based on the spatial positions.

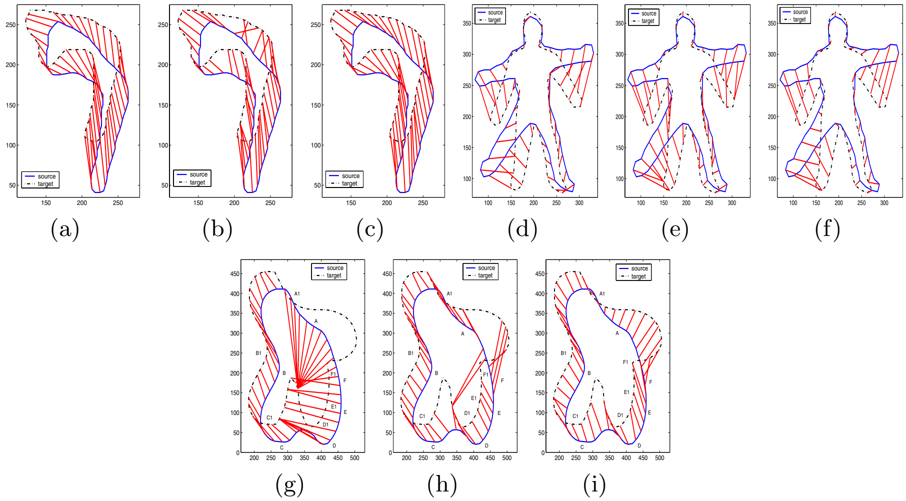
In all experiments reported in this paper, the size of the shape context histogram is 5×12 and the increasing coefficient in (7) is taken as 2.5. When performing SC and the BP matching algorithm, the iteration times are both manually set as 30. We report three experimental results in Figure 4.

In Figure 4(a) and 4(d), we can see that the results obtained by standard DP approach are very good except that the first and last several points miss to be matched. In Figure 4(g), the source contour and target contour have different local non-rigid deformations except that the two segments 'ABC' and 'DEF' in source contour are equal to 'A1B1C1' and 'd1E1F1' in target contour, respectively. Although the correspondence are all order preserving, there are a lot of points mapped to a same target point, resulting in that a lot of continuous points in the target miss to be matched.

In Figure 4(b) and Figure 4(h), there exist evident cross correspondences. In Figure 4(e), some many-to-one correspondences are generated.

Figure 4(c), 4(f) and 4(i) demonstrate the results which are obtained by the standard BP matching algorithm introduced in Section 3. We can see that the results are better than those obtained by standard DP approach and the SC approach. Globally, good local matching is achieved. Actually, in our BP matching framework, we explicitly penalize the two cases: i.e. cross correspondences and many-to-one correspondences. We also calculate the cost of $V_{ij}(f_i, f_j)$ in a circular way. This guarantees that the first and the last several indexed points in target contour can also be matched correctly.

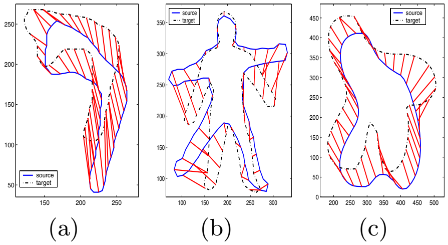
In Figure 5, we illustrate the results obtained by the fast BP (FBP) matching algorithm. The iteration times are also set as 30. As can be seen, the results are very similar to those obtained by the standard BP.



(a) (b) (c) (d) (e) (f)

(g) (h) (i)

Fig. 4. Three examples. (a), (d) and (g) demonstrate the results matched by dynamic programming; (b), (e) and (h) show those by soft correspondence, and (c), (f) and (i) show those by our approach. Contours in (a) are extracted manually from two frames in a diving video, contours in (d) correspond to two postures of two persons, while contours in (g) are drawn by hand.



(a) (b) (c)

Fig. 5. (a), (b) and (c): Results by the fast matching algorithm

Table 1. Computation time (100ms) (CPU: Intel Pentium 2.4GHz; RAM: 512M)

NN	60	90	120	150	180	210	240	270	300	330	360
DP	0.01	0.02	0.03	0.04	0.06	0.09	0.15	0.24	0.35	0.43	0.70
SC	0.02	0.05	0.09	0.17	0.31	0.47	0.62	0.78	0.94	1.05	1.41
SBP	0.83	2.80	6.47	12.3	20.9	33.0	48.6	69.0	94.0	124.	161.
FBP	0.03	0.09	0.26	0.53	0.81	1.12	1.40	1.93	2.58	3.09	3.63

To analyze computation complexity, we perform the algorithms with different number of nodes (NN). Table 1 illustrates the computation time, i.e. the average time of 10 tests, which does not contain the time spending on calculating the local shape features. When using the standard BP (SBP), we store all the $V_{ij}(f_i, f_j)$

in a table to improve the computation speed. We can see that the computation time is drastically reduced when using FBP approach.

6 Conclusion

This paper introduces a new method for contour matching builded on belief propagation. Each of the points, which are used to define the source contour, is associated with a graph node to construct an undirected loopy graph. As observation information, the distance between source point and target point to be matched is measured by the shape context descriptor and curvature information. When computing the discontinuity cost of one message, we explicitly penalize two cases of incorrect correspondences: cross correspondence and many-to-one correspondence. Finally, the messages are transferred iteratively on this graph and a geometrical mapping can be obtained by minimizing the energy function and maximizing a posterior for each node.

The standard belief propagation for contour matching is time consuming. To reduce computation complexity, this paper introduces a fast algorithm and satisfactory matching results are achieved.

In the further, we would like to use belief propagation to match segment-to-segment contour matching.

Acknowledgements

This study is carried out as a part of “R&D promotion scheme funding international joint research” promoted by NICT (National Institute of Information and Communications Technology) of Japan.

References

1. Veltkamp, R. C., Hagedoorn, M.: State of the art in shape matching. Tech. Rep. UU-CS-1999-27, Utrecht University, Holand (1999)
2. Grimson, E.: Object Recognition by Computer: The Role of Geometric Constraints. MIT Press, Cambridge, MA, USA (1990)
3. Belongie, S. Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. Pattern Analysis and Machine Intelligence, **24** (2002) 509–522
4. Chui, H., Rangarajan, A.: A new algorithm for non-rigid point matching. In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Hilton Head Island, SC, USA, **2** (2000) 44–51
5. Rangarajan, A., Coughlan, J., Yuille, A. L.: A bayesian network framework for relational shape matching. In: Proc. of IEEE Int. Conf. on Computer Vision, Paris, France (2003) 671–678
6. Coughlan, J. M., Ferreira, S. J.: Finding deformable shapes using loopy belief propagation. In: Proc. of European Conf. on Computer Vision, Copenhagen, Denmark (2002) 453–468
7. Milios, E., Petrakis, E.: Shape retrieval based on dynamic programming. IEEE Trans. on Image Processing, **9** (2000) 141–147

8. Scott, C., Nowak, R.: Robust contour matching via the order preserving assignment problem. http://www.stat.rice.edu/ cscott/pubs/copap.pdf (2004)
9. Loncaric, S.: A survey of shape analysis techniques. Pattern Recognition, **31** (1998) 983–1001
10. Luo, B., Hancock, E. R.: Structural graph matching using the em algorithm and singular value decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence, **23** (2001) 1120-1136
11. Srisuk, S., Tamsri, M., Fooprateepsiri, R., et al.: A new shape matching measure for nonlinear distorted object recognition. In: Proc. of Digital Image Computing: Techniques and Applications, Sydney, Australia (2003) 339–348
12. Frenkel, M., Basri, R.: Curve matching using the fast marching method, In: Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, Lisbon, Portugal (2003) 35–51
13. Sebastian, T. B, Klein, P. N., Kimia, B. B.: Recognition of shapes by editing their shock graphs. Pattern Analysis and Machine Intelligence, **26** (2004) 550–571
14. Grauman, K., Darrell, T.: Fast contour matching using approximate earth mover's distance. In: CVPR, Washington DC, USA (2004) 220–227
15. Weiss, Y., Freeman, W. T.: On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. IEEE Trans. on Information Theory, **47** (2001) 736–744
16. Thayananthan, A., Stenger, B., Torr, P. H. S., et al.: Shape context and chamfer matching in cluttered scenes. In: CVPR, Madison Wisconsin (2003) 127–133
17. Felzenszwalb, P. F., Huttenlocher, D. P.: Efficient belief propagation for early vision. In: CVPR, Washington DC, USA (2004) 261–268
18. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publisher, San Francisco, CA, USA (1988)

Key Frame-Based Activity Representation Using Antieigenvalues*

Naresh P. Cuntoor and Rama Chellappa

Center for Automation Research, University of Maryland,
College Park, MD, 20742, USA
{cuntoor, rama}@cfar.umd.edu
http://www.cfar.umd.edu/users/cuntoor

Abstract. Many activities may be characterized by a sequence of key frames that are related to important changes in motion rather than dominant characteristics that persist over a long sequence of frames. To detect such changes, we define a transformation operator at every time instant, which relates the past to the future states. One of the useful quantities associated with numerical range of an operator is the eigenvalue. In the literature, eigenvalue-based approaches have been studied extensively for many modeling tasks. These rely on gross properties of the data and are not suitable to detect subtle changes. We propose an antieigenvalue - based measure to detect key frames. Antieigenvalues depend critically on the turning of the operator, whereas eigenvalues represent the amount of dilation along the eigenvector directions aligned with the direction of maximum variance. We demonstrate its application to activity modeling and recognition using two datasets: a motion capture dataset and the UCF human action dataset.

1 Introduction

The scope of modeling human activities has expanded from recognizing simple activities such as walking, running and making hand gestures, to more complex ones that involve an underlying structure. While statistical techniques have been applied in the case of simple activities ([1],[2]), primitive - based approaches that rely on domain knowledge have been proposed for complex ones ([3], [4]). We attempt to provide an unsupervised key frame based representation for human activities by focusing on changes in motion properties rather than a sequence of dominant features that form primitives.

In many activities, the relevant information is contained in a few key frames. These frames may be significant due to certain changes in the data, such as direction, speed and deviation from a known behavior. As an illustration, consider the trajectory traced by a hand when opening the door. The shape of the trajectory depends on the person opening the door, the initial position of the

* This work was sponsored by the Advanced Research and Development Activity, a U.S. Government entity which sponsors and promotes research of import to the intelligence community.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 499–508, 2006.

hand, the camera's viewing direction, etc. Modeling these variations is neither easy nor relevant for the activity of opening. The opening action occurs within a few frames when the hand makes contact with the door. The sequence of key frames - extending the hand, grabbing the handle and opening the door - is a sufficient representation. Similarly, we may say that walking is a sequence of events or key stances including the rest stance when the feet are closest to each other and the swing stance when the feet are maximally apart. Jogging may be represented by a similar set of freeze frames, but the changes from frame to frame are different from those of walking.

The theory of antieigenvalues is based on changes in the data. It is sensitive to how much a data vector is turned from a known direction, rather than the direction of persistence [5]. On the other hand, eigenvectors represent the direction of maximum spread of the data and the eigenvalues are proportional to the amount of dilation. We propose an antieigenvalue-based approach for detecting key frames by investigating properties of operators that transform past states to observed future states.

The paper is organized as follows. Section 2 motivates the key-frame based representation for activities. Section 3 gives a brief overview of antieigenvalue theory. Section 4 describes the proposed approach. Section 5 demonstrates the proposed method using two datasets: the MOCAP database and the UCF human action database. Section 6 concludes the paper.

1.1 Prior Work

Aggarwal and Cai [6] present a comprehensive review of human motion and human activities. Ivanov and Bobick [7] propose a two-step procedure where primitives are modeled using HMMs and a sequence of primitives is parsed using stochastic grammar. Hamid et al. [3] present a dynamic Bayesian network framework for tracking and recognizing complex multi agent activities. Vaswani et al. regard a sequence of moving points engaged in the activity as a shape using Kendall's shape space theory [8]. Nevatia et al. [9] present an Event Representation Language (ERL) that captures the ontological structure of activities using events. Rao et al. [10] detect *dynamic instants*, which are defined as points of maximum curvature along a trajectory. Parameswaran and Chellappa [11] compute view invariant representations for human actions in both 2D and 3D. State-space approaches have been used by many researchers. For example, Brand et al. [1] use coupled HMMs to model human actions that involve multiple parts such as hands and the head. Eigenvalue (and singular value)-based methods have been used extensively in many modeling tasks including face, gait and activities ([12], [13], [14]).

2 Key Frame Representation

As we argued through examples of opening a door, walking, etc., many activities can be represented using key frames instead of the entire video sequence. Generally, there are three ways to decide on what constitutes a key frame. We may use domain knowledge in a top-down fashion. It requires an extensive model for the activity, which may be tedious. It relies on our ability to detect the key frames across variations in the data that occur due to structural changes and noise [3]. We may hypothesize that the important characteristics of the activity are present in the persistent and dominant frames [14]. This makes it difficult to detect subtle changes, since it may be difficult to distinguish them from noise. We may look for key frames that are a result of certain changes in the data. In other words, changes in the activity may be more useful than the absolute values of a dominant feature in representing the activity. We present an unsupervised approach for detecting key frames based on changes in the data.

Let the past state vector $\mathbf{x}_-$ be transformed by an operator A_t to a future state vector $\mathbf{x}_+$. If motion properties do not change appreciably, then $\mathbf{x}_+$ may be related to $\mathbf{x}_-$ by an identity transformation modulo translation. Such a transformation may be less interesting compared to the case where A_t turns the state $\mathbf{x}_-$. We show how antieigenvalues may be used to detect such changes and to identify the key frames. In contrast, eigenvalues are tuned to detecting identity-like transformations. It is important to point out that these quantities are of intrinsic interest in their own right. As the term denotes, however, it may be easier to gain an insight into antieigenvalues by contrasting with eigenvalues and eigenvectors of the operator.

The motion trajectories are associated with two quantities: the antieigenvalue sequence, which is the sequence of antieigenvalues for the operator A_t for every time t, and the location of the key frames detected using minima in the average antieigenvalue sequence. Both the extent of change as given by the antieigenvalues and the location of key frame are useful for recognition. If viewing conditions change, we may expect the time instants of occurrence of key frames to be more useful since the extent of change depends on viewing direction. On the other hand, if the viewing direction is fixed, antieigenvalues may be used in comparing two activities. We illustrate both these cases in our experiments.

3 Mathematical Preliminaries: Antieigenvalues

We present a brief description of antieigenvalues before discussing its application. A detailed discussion of antieigenvalues may be found in [5] or [15].

For a square matrix A, a non-zero vector $\mathbf{x}$ is said to be an eigenvector if $A\mathbf{x} = \lambda\mathbf{x}$, and λ is called the eigenvalue. Equivalently, we may state the condition as $\cos\theta = 1$, where θ is the angle between $\mathbf{x}$ and $A\mathbf{x}$. Geometrically, we may think of eigenvectors as those that dilate A but do not turn at all. The eigenvalues represent the amount of dilation. On the other hand, antieigenvectors are critical to the turning of A. Instead of seeking $\cos\theta = 1$ or $\theta = 0$, antieigenvectors minimize $\cos\theta$, or equivalently, maximize θ. The n^{th} antieigenvalue is defined variationally [5] as

$$\mu_n(A) = \inf_{A\mathbf{x_n} \neq 0} \frac{\Re\langle A\mathbf{x_n}, \mathbf{x_n}\rangle}{\|A\mathbf{x_n}\|\|\mathbf{x_n}\|}, \tag{1}$$

where the n^{th} antieigenvector $\mathbf{x_n} \perp \{\mathbf{x_1}, \ldots, \mathbf{x_{n-1}}\}$. It has been shown [15] that all antieigenvectors for 2×2 matrices are of the form

$$\mathbf{x} = \left(\frac{\pm\sqrt{\lambda_j}}{\sqrt{\lambda_i + \lambda_j}}, \frac{\sqrt{\lambda_i}}{\sqrt{\lambda_i + \lambda_j}} \right), \tag{2}$$

where i, j index all eigenvalues. For example, let

$$A = \begin{pmatrix} 9 & 0 \\ 0 & 16 \end{pmatrix} \tag{3}$$

The eigenvalues of A are $\lambda = 9, 16$. Using (2), the first antieigenvector is $\mathbf{x_1} = (\frac{-4}{5}, \frac{3}{5})$. The antieigenvalue may be calculated by substituting the value of x_1 in (1). The first antieigenvalue is $\mu_1(S) = \frac{\langle A\mathbf{x_1}, \mathbf{x_1} \rangle}{\|A\mathbf{x_1}\|} = 0.96$ The second antieigenvector is $\mathbf{x_2} = (\frac{3}{5}, \frac{4}{5})$ and the corresponding antieigenvalue is 0.97.

The first total antieigenvalue is defined as $|\mu_1(A)| = \inf_{A\mathbf{x} \neq 0} \frac{|\langle A\mathbf{x}, \mathbf{x} \rangle|}{\|A\mathbf{x}\|\|\mathbf{x}\|}$. The higher total antieigenvalues are similarly defined.

The total antieigenvalues for matrices of size greater than 2×2 may be calculated as follows (theorems 2.1 and 2.2 in [5]). Let A be a normal operator with eigenvalues $\lambda_i = \beta_i + i\delta_i$, $i = 1, \ldots, n$. Then the first total antieigenvalue is either 1 or the smallest number in the set of values

$$G = \{ \frac{\sqrt{(\beta_i|\lambda_j| + \beta_j|\lambda_i|)^2 + (\delta_i|\lambda_j| + \delta_j|\lambda_i|)^2}}{(|\lambda_i| + |\lambda_j|)\sqrt{|\lambda_i||\lambda_j|}}, \tag{4}$$

where $i \neq j, 1 \leq i \leq n, 1 \leq j \leq n\}$. If $|\mu_1(A)| = 1$, then the first total antieigenvector is $\mathbf{z_1} = (z_1, z_2, \ldots, z_n)$ with $|z_j| = 1$ for some j and all other $z_i = 0$. If $|\mu_1(A)|$ is one of the values in G, then the components of $\mathbf{z_1}$ satisfy $|z_i|^2 = \frac{|\lambda_j|}{|\lambda_i| + |\lambda_j|}, |z_j|^2 = \frac{|\lambda_i|}{|\lambda_i| + |\lambda_j|}$, all all other $z_k = 0$. Further, all higher total antieigenvectors take their value from the set G and the corresponding higher total antieigenvectors possess the same component structure as the first total antieigenvector.

4 Key Frame Detection Using Antieigenvalues

In this section, we describe the proposed antieigenvalue-based key frame detection procedure. The key frames are used to compare two activities.

4.1 Feature Selection

We obtain trajectories of the moving object and compute its apparent velocities. The tracking procedure for the different datasets is outlined in section 5. The state of a moving object is said to be the tuple $(x(t), y(t), \dot{x}(t), \dot{y}(t))$, where $(x(t), y(t))$ represents the instantaneous position. We assume that the state undergoes certain important changes at the key frames. We are interested in detecting these changes, rather than modeling the entire sequence of frames. Let

$A_t : H \to H$ be an operator that relates the past state $\mathbf{x}(t_-)$ into the future state $\mathbf{x}(t_+)$, where H is the Hilbert space domain. There are two estimation tasks here. We need to estimate the past and future states $\mathbf{x}(t_-)$ and $\mathbf{x}(t_+)$. For robust estimation, we assume that the state of the system remains constant for a short interval of time. The other estimation tasks involves optimizing the parameters of the operator A_t. If there is no change in the state from t_- to t_+, we may expect A to be the identity matrix (modulo translation).

4.2 Computing the Transformation Operator

We assume that the speed remains approximately constant for W frames. The value of W depends on the type of data. For instance, it may be reasonable to assume $W = 25$ or 1 second in far field surveillance data. On the other hand, we may assume $W = 3$ or 0.1 second for short-term human actions (e.g. opening the door, picking up an object, etc.) performed in an office environment. Using W frames of the data, we estimate the state variables $\mathbf{x}(t_-)$ and $\mathbf{x}(t_+)$. Assume that the two states are related by a linear transformation, i.e., $x(t_+) = A_t x(t_-)$. We estimate the parameters of the operator A_t using least squares technique and W frames each for $\mathbf{x}(t_-)$ and $\mathbf{x}(t_+)$.

For two vectors $\mathbf{x}, \mathbf{b} \in \mathcal{R}^n$, let A be the transformation operator such that $A\mathbf{x} = \mathbf{b}$, where $A = [a_{ij}]$, $i, j = 1, 2, \ldots, n$. This can be rewritten as $X\mathbf{a} = \mathbf{b}$, where $\mathbf{a} = (a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, \ldots, a_{2n}, \ldots, a_{n1} \ldots, a_{nn})$ and X is a matrix that consists of rows of the form $(0, 0x_1, x_2, \ldots, x_n, 0, 0, \ldots, 0)$. Suppose $A\mathbf{x} = \mathbf{b}$ holds for W vector pairs $(\mathbf{x_1}, \mathbf{b_1}), \ldots, (\mathbf{x_W}, \mathbf{b_W})$, we can write

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_W \end{pmatrix} \mathbf{a} = \begin{pmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \vdots \\ \mathbf{b_W} \end{pmatrix} \tag{5}$$

We use least squares technique to solve for $\mathbf{a}$ in (5) and recompose the vector $\mathbf{a}$ into the matrix A.

4.3 Numerical Range of the Operator

The numerical range of an operator A is defined as the set $W(A) = \{\langle A\mathbf{x}, \mathbf{x}\rangle, \mathbf{x} \in H, \|\mathbf{x}\| = 1\}$, where H is the Hilbert space. For example, consider an operator defined by the matrix $A = \begin{pmatrix} 0 \, 1 \\ 0 \, 0 \end{pmatrix}$. Let $\mathbf{x} = (p, q)$. For simplicity, assume $\|\mathbf{x}\| = |p|^2 + |q|^2 = 1$. Then $A\mathbf{x} = (q, 0)$ and $\langle A\mathbf{x}, \mathbf{x}\rangle = qp$. A simple calculation shows that $W(A) = \{\mathbf{x} = (p, q) : |p|^2 + |q|^2 \leq \frac{1}{2}\}$ or the half disk. Closely related to the numerical range, we can define the angle of the operator $\cos A$ and the antieigenvalues of the operator A as discussed in section 3.

4.4 Choosing Key Frames

We compute antieigenvalues of A_t using (4) and use the mean antieigenvalue as a measure of relative significance of the frame in representing the activity.

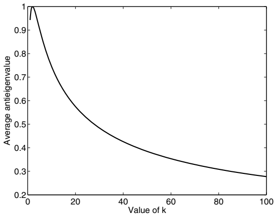


Fig. 1. Average antieigenvalue for A in (6) as a function of increasing k, the change in velocity

A small value of the mean antieigenvalue indicates that the minimum $\cos A_t$ is small or that the turning angle is large. This indicates a larger relative change in the state vector and hence significant for representing the activity. We illustrate the use of antieigenvalues in detecting key frames through a few examples in the 1-D case. The state of the moving object is the pair $(x(t), \dot{x}(t))$. Suppose the transformation operator is given by

$$A = \begin{pmatrix} 2 & 0 \\ 0 & k \end{pmatrix}. \tag{6}$$

For differing values of k, this means that the change in the state of the object is due to a changing speed, while the position remains constant (modulo translation). Figure 1 shows the variation of the average antieigenvalue as the value of k is increased. We observe that, as expected, the average antieigenvalue varies inversely as the extent of change in the state of the moving object.

4.5 Matching Two Sequences

We compute the similarity score between two video sequences by comparing the sequences of key frames. Clearly, an activity need not be repeated with the same timing scale from one instantiation to the next and the location of key frames may change slightly. To allow for non-linear time normalization while matching, we use dynamic time warping (DTW) [16]. The similarity score is computed by traversing the warping path, which gives the correspondence of the frames in the reference and probe sequences.

To place the proposed approach in context, we compare this to the eigenvalue based methods. Various approaches in the literature have used eigenvalue-based ideas to model activities in two main ways: for pre-processing or filtering the data and for extracting the dominant characteristics for representation. The basic hypothesis in all these approaches is that the dominant characteristics of the signal are important. Also, the main characteristics are assumed to be highly structured and stationary. In such a setting, the eigenvectors capture the dominant characteristics and the eigenvalues represent the relative contribution of the eigenvectors for representation. For example, eigenfaces capture the dominant characteristics for face recognition [12]. By reconstructing the signal using the top few eigenvectors, it induces a smoothing operation on the original signal

[13]. Zhong et al. [14] use this idea for activity classification where they cluster the sequence of frames into prototype classes.

4.6 Algorithm Overview

- Pre-processing: Extract object trajectory from video and smooth it.
- For every time t, compute the state $\mathbf{x}(t) = (x(t), y(t), \dot{x}(t), \dot{y}(t))$. For computing $\dot{x}(t), \dot{y}(t))$, we use finite differencing over W frames of data.
- Compute the least squares estimate of the operator $A_t : \mathbf{x}(t_-) \rightarrow \mathbf{x}(t_+)$.
- Compute the antieigenvalues of the operator A_t using (4). Compute its mean.
- Recognition: compare the key frames detected from the average antieigenvalue sequence for the training using DTW.

5 Experiments

We demonstrate our approach to activity recognition using the MOCAP action dataset and the UCF human action dataset.

5.1 Motion Capture (MOCAP) Dataset

The MOCAP dataset available from Credo Interactive Inc. and Carnegie Mellon University consists of motion capture data of subjects performing different activities including different kinds of walking, jogging, sitting and crawling. The system tracks 53 joint locations and the tracks are stored in the bvh format. Since not all the 53 points are relevant, we use only a few of the trajectories. For example, trajectories of the different fingers and toes may not be as informative as the location of the arms, legs or hip for activities such as walking or sitting. We choose 5 regions of the 53 locations to demonstrate activity classification. This dataset allows us to test the efficacy of the proposed method in the absence of noise and errors due to low-level issues. There are 9 activities in the dataset and approximately 75 sets of observation overall. The tracks for an activity such as walking consists of multiple cycles of the activity. We divide the sequence into individual walking cycles and treat each half-cycle as an observation. Half-cycle refers to the part of the walking cycle starting from the standing pose, right (or left) leg forward, reaching the swing pose, and withdrawing the right (or left) leg to the standing pose. The number of observations is increased to 365 by treating similar trajectories of nearby locations as multiple samples, i.e., 2 locations near the abdomen are treated as multiple samples of the same location. To ensure that there is no bias due to the displacement, we use mean-subtracted trajectories for all locations.

We compute the state vector for every time instant and estimate the transformation operator A_t as described in section 4. We compute the antieigenvalue and use its mean as a signature for the activity. The antieigenvalue sequences are matched using DTW. All the activities were correctly recognized. Table 1 summarizes the activities that were the closest matches following the top match. We

Table 1. MOCAP dataset: Closest-matching activities based on comparing event probability sequences. All activities were correctly recognized. Table shows the matches following the top match.

Test activity	Match #2	Match #3
Blind-walk	Normal walk	Normal walk
Prowl-walk	Jog	Exaggerated walk
Broom	Sit	Exaggerated walk
Crawl	Broom	Sit
Exaggerated walk	Sad walk	Normal walk
Jog	Jog2	Normal walk
Sit	Sit1	Neutral
Normal walk	Normal walk	Sad Walk
Sad walk	Exaggerated walk	Normal walk

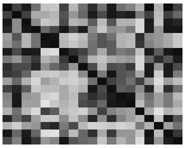


Fig. 2. Confusion matrix for activities in the MOCAP dataset

observed that the different types of walking resembled each other while the similarity scores corresponding to *sitting*, *sweeping with a broom* were significantly larger. Figure 2 shows the confusion matrix across all activities. It may not be straightforward to associate a physical meaning to the detected key frames for activities such as walking, etc. other than saying a key frame was detected at the stance when the feet are maximally apart, and so on. In the UCF action dataset described below, the key frames are more readily apparent.

5.2 UCF Human Actions Dataset

The UCF dataset consists of 60 trajectories of common activities. We divide these into 7 classes: open door, pick up, put down, close door,erase board, pour water into cup and pick up object and put down elsewhere. The hand trajectories are obtained after initialization using a skin detection technique. The resulting trajectories are smoothed out using anisotropic diffusion. A detailed description of the dataset, tracking and smoothing operations are available in [10].

The average antieigenvalue sequence was computed as outlined in section 4.6. The key frames were identified by finding the minima in the average antieigenvalues. Figure 3 shows the key frames identified for some of the activity trajectories. The dots marked along the trajectory denote the key frames detected along the trajectory. Figure 3(a) shows the key frames for opening a door. In figure 3(b), the trajectory for picking up an object from the desk and putting it on the floor shows two key frames detected, one of which is the result of a sharp change in

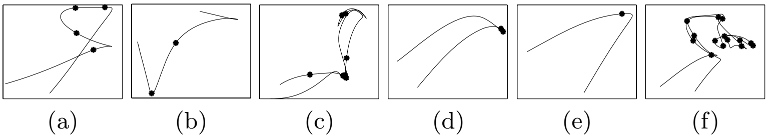


Fig. 3. Sample trajectories from UCF dataset showing key frames detected

direction and the other a gradual change. The second sharp change is not detected due to boundary effects. In the case of erasing a white board, we observe a key frame when the eraser is picked up, and several key frames at the left side of the erasing back-and-forth action of the hand (figure 3(c). This means that each back and forth action of the hand may be considered as the past and future states separated by the key frames. Figures 3(d) and (e) show trajectories of picking up objects. They each have one key frame detected at approximately the instant the object is picked up. Figure 3(f) shows the trajectory of a random action. The lack of structure in the data is reflected by a large number of changes leading to the detection of several key frames.

Comparison with the UCF method[10]: Rao et al. treat activities as a sequence of *dynamic instants* that are defined as the points of maximum curvature along the trajectory [10]. The key frames in the proposed approach are detected based on changes in the data including changes in direction and changes in speed. The comparison of recognition rates are given in figure 4.

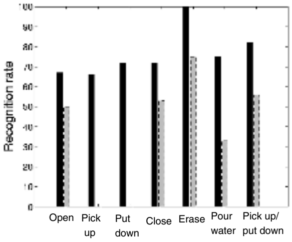


Fig. 4. UCF dataset: Comparing recognition rates. Solid black bar represents proposed method, dashed gray bar are the rates reported in [10].

6 Summary

We have presented a key frame based activity representation using the largely unexplored theory of antieigenvalues. We have argued that key frames should be related to changes in the data, rather than dominant, persistent properties. This allows a natural way to detect both subtle and sudden changes, which are often more interesting than the portions of the data that are normally observed. As part of future work, we will investigate the measures to compare antieigenvalues. It may be useful to obtain more efficient ways of calculating antieigenvalues.

References

1. Brand, M., Oliver, N., Pentland, A.: Coupled hidden markov models for complex action recognition. In: Proc. CVPR. (1996) 949–999
2. Wilson, A.D., Bobick, A.F.: Parametric hidden markov models for gesture recognition. IEEE Trans. PAMI **21** (1999) 884–900
3. Hamid, R., Huang, Y., Essa, I.: Argmode - activity recognition using graphical models. In: Proc. CVPR. Volume 4., Madison, WI, USA (2003) 38–43
4. Ghanem, N., Dementhon, D., Doermann, D., Davis, L.: Representation and recognition of events in surveillance video using petri nets. In: Proc. IEEE Workshop on Event Mining. (2004)
5. Gustafson, K.: Antieigenvalues. Linear Algebra and Appln. **208** (1994) 437–454
6. Aggarwal, J., Cai, Q.: Human motion analysis:a review. Computer Vision and Image Understanding **73** (1999) 428–440
7. Ivanov, Y.A., Bobick, A.F.: Recognition of visual activities and interactions by stochastic parsing. IEEE Trans. PAMI **23** (2000) 852–872
8. Vaswani, N., Chowdhury, A.R., Chellappa, R.: Activity recognition using the dynamics of the configuration of interacting objects. In: Proc. CVPR. (2003)
9. Nevatia, R., Zhao, T., Hongeng, S.: Hierarchical language-based representation of events in video streams. In: Proc. IEEE Workshop on Event Mining. (2003)
10. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. International J. Comput. Vision **63** (1989) 257–285
11. Parameswaran, V., Chellappa, R.: View invariants for human action recognition. In: Proc. CVPR. (2003)
12. Turk, M.A., Pentland, A.: Face recognition using eigenfaces. In: Proc.CVPR. (1991)
13. Kale, A., Rajagopalan, A.N., Sundaresan, A., Cuntoor, N., Roy-Chowdhury, A.K., Kruger, V., Chellappa, R.: Identification of humans using gait. IEEE Trans. Im. Processing. (2004) 1163–1173
14. H. Zhong, J.S., Visontai, M.: Detecting unusual activity in video. In: Proc. CVPR. (2004) 819–826
15. Gustafson, K., Seddinghin, M.: Antieigenvalue bounds. J. Math. Anal. and Appln. **143** (1989) 327–340
16. Juang, B.H.: On the hidden markov model and dynamic time warping for speech recognition - a unified view. Technical Journal **63** (1984) 1213–1243

Fast Image Replacement Using Multi-resolution Approach

Chih-Wei Fang and Jenn-Jier James Lien

Robotics Laboratory, Dept. of Computer Science and Information Engineering,
National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan, Taiwan
{nat, jjlien}@csie.ncku.edu.tw
http://robotics.csie.ncku.edu.tw

Abstract. We developed a system including two modules: the texture analysis module and the texture synthesis module. The analysis module is capable of analyzing an input image and performing the training process by using this image data. According to the training non-periodic or periodic pattern, we use different sampling methods to have different amount of patches in order to reduce the emergences of the seams of the output synthesized image. In addition, the properties of principal component analysis (PCA) are used to reduce the dimensions of the data representation and to recombine the appearance of the features (i.e. eigenvectors). Then the vector quantization (VQ) algorithm is employed to reduce the time spent on matching comparison. For the synthesis module, the training data is used to synthesize a large output texture, or is employed to replace the removed regions of an image. The multi-resolution approach is applied to accelerate the procedure of our algorithm: the down-sampling step is the training process and the up-sampling step is in the order of reconstructing (or synthesizing) the large removed region without needing to assign initial random values or approximate values. Therefore, our system can rapidly obtain a high image quality and promising result.

1 Introduction

Photographs sometimes include unwanted objects. Conversely, it is sometimes desirable to replace an existing object in a photograph by a new object. Although techniques, which enable the unwanted object to be removed and replaced by a new object, do exist a problem frequently in that the shapes and the sizes of the two objects differ, and hence a gap is apparent in the reconstructed image. Consequently, the present study develops an approach to fill in the gap left in an image when an unwanted object is removed.

Most existing algorithms are based on obtaining a minimum difference between the synthesizing patches. These represent having the minimum parallax in vision. In some approaches, the continuous structures are important and the discontinuous features are rough. However, these methods are very slow and result in a loss of definition in the reconstructed images. Clearly then, there exists a requirement to develop image replacement algorithms. Accordingly, the present study develops a method in which an approximation is used to establish suitable patches with which to fill the gap in the image. In the developed process, additional techniques are adopted in order to accelerate the process without losing its detail features.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 509 – 520, 2006.

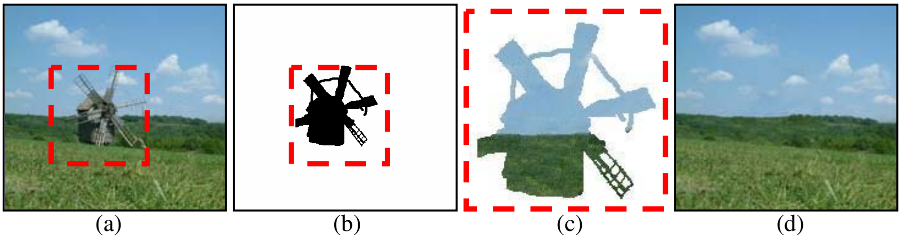


Fig. 1. (a) Input image. (b) Inverse matte that defines the removed (or replaced) region. (c) The result of the synthetic region. The result is shown by zooming in the replaced region. (d) Output synthesized image.

2 Related Work

Several researchers have developed gap-filling methods based on an examination of the neighborhood pixels [1], [12], [13], or using dynamic programming (DP) [4], [8] to find the minimum error for cutting. However, these methods have a gradual change in two different sources. Hence, sharp-pointed edges become blurred. Recent studies have presented the use of analysis and sampling techniques for the regular or near-regular patterns, and have then used tiles to synthesize the image [5], [10]. These approaches commence by identifying the features or structures. These are then arranged as skeletons images, and the texture is wrapped onto these skeletons [10], [14], [16]. Such methods are suitable for the regular or near-regular patterns. They can solve the discontinuous condition. But these methods are inapplicable to non-structured and non-regular images.

In an alternative approach, an inverse matte is used to preserve the necessary background while removing the unnecessary region [2], [7]. In nature, the noises are Gaussian signals. These noises spread everywhere and cause a photograph to be truer and more pellucid. In order to merge the fragments more smoothly with no seam, Drori et al. [2] used a Gaussian mask to blend the various fragments. This approach successfully creates a flawless image. However, the results tend to be more blurred and some detailed feature information is lost.

Some studies use patch-based approaches to synthesize image, but these approaches may make the structure of reconstructed image discontinuous [9], [11]. Similar patches are then developed to fill in the resulting holes. In this technique, weighting information is used to blend patches that will lose the detailed features.

It has been shown that multi-resolution approach provides a reasonable technique for obtaining an initial value or for developing an approximate result, called push-pull [2], [3], [6], [15]. The approximate result can then be used as the basis to develop a superior result. Although this method can yield reasonable results, it involves the use of repeated reconstruction procedures, with the result that errors can occur. Furthermore, these errors can be compounded in the subsequent reconstruction steps.

To solve above existing problems, this paper proposes two modules: texture analysis (training) module and texture synthesis module. The texture analysis module is capable of analyzing input images (or textures) and training by using these data (Section 3). The synthesis module can rapidly synthesize a large image (or texture) based on these training data (Section 4). The synthesis process performs on a real-time basis. In addition, the synthesis process is then modified by applying the multi-resolution approach to the image replacement process by initially assigning meaningful definition values instead of giving random values or approximate values (Section 5). This approach also accelerates the image replacement process and is capable of handling the large removed region. Section 6 presents the current experimental results by evaluating time cost of the training and reconstruction (or synthesis) processes, and analyzing our developed approaches. Finally, Section 7 presents the conclusion.

3 Texture Analysis (Training) Module

The output big image is synthesized based on the small single input image (texture). The input image is divided into several patches, which are investigated in the subsequent analysis process. The important data of the input image are selected for

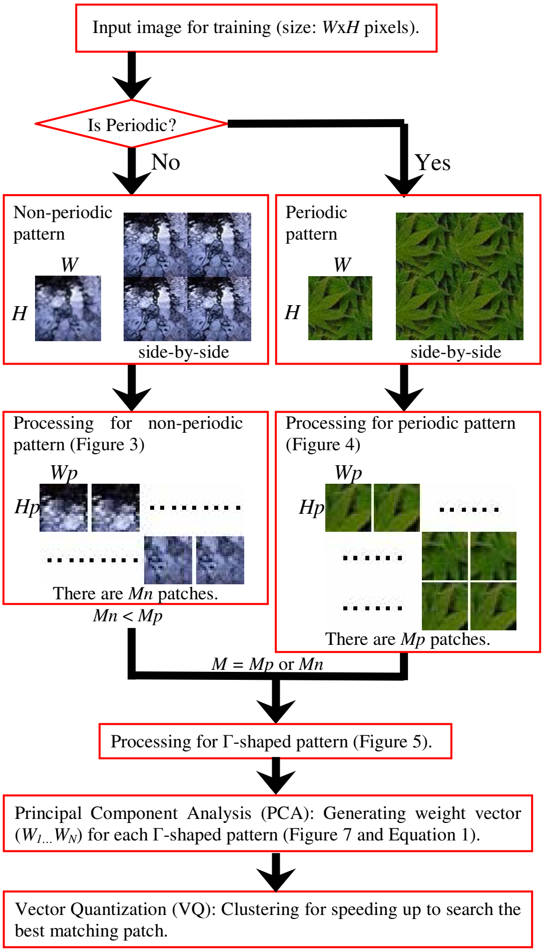

Input image for training (size: *W*x*H* pixels).
Is Periodic?
No
Yes
Non-periodic pattern
W
H
side-by-side
Periodic pattern
W
H
side-by-side
Processing for non-periodic pattern (Figure 3)
Wp
Hp
There are *Mn* patches.
Mn < *Mp*
Processing for periodic pattern (Figure 4)
Wp
Hp
There are *Mp* patches.
M = *Mp* or *Mn*
Processing for Γ-shaped pattern (Figure 5).
Principal Component Analysis (PCA): Generating weight vector ($W_1 ... W_N$) for each Γ-shaped pattern (Figure 7 and Equation 1).
Vector Quantization (VQ): Clustering for speeding up to search the best matching patch.

Fig. 2. Framework of the texture analysis (training) module

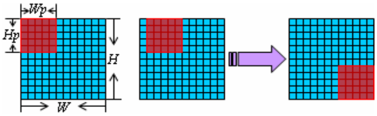


Fig. 3. Processing for non-periodic pattern. A pixel-by-pixel shifting method is used to divide the input texture into M patches, where M = (W–Wp+1)×(H–Hp+1), in which W is the width of the input image (or texture), H is the height of the input image, Wp is the width of the patch, and Hp is the height of the patch.

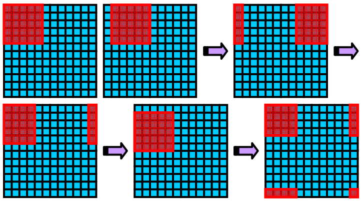


Fig. 4. Processing for periodic pattern. The pattern is divided into M patches, where M=W×H. When the patch reaches the boundary of the input image, in order to form a complete patch, additional pixels are supplied from the opposite border.

training purposes. Principal component analysis (PCA) is employed to reduce the dimensions of the data, and vector quantization (VQ) is adopted to reduce the time required for comparison. Figure 2 illustrates the system framework extending from the input image to the output image.

3.1 Processing for Non-periodic and Periodic Patterns

The training data is obtained by cropped patches (or windows) of pre-defined size (*Wp*x*Hp* pixels) from the original input image (*W*x*H* pixels). Two different patch-dividing schemes are employed depending on whether the image has a non-periodic pattern or a periodic pattern, as shown in Figure 3 and Figure 4, respectively. Periodic patterns have continuous veins between two equivalent patterns when positioned side by side (see Figure 2). Non-periodic patterns do not have this property. When non-periodic patterns are placed side by side, there is a visible discontinuity seam between the textures (also see Figure 2). Therefore, periodic patterns can be divided into more patches than non-periodic patterns. For example, an input image of size 64x64 pixels can be divided into 4096 patches of size 32x32 pixels if the pattern is periodic, but can only be divided into 1089 patches if the pattern is non-periodic.

3.2 Processing for Γ-Shaped Pattern

Taking the whole pieces of the patch as training data may lead to an overestimation of the underlying structure of the patch and will certainly increase the length of the training time. In addition, searching the matching patches by considering their whole contents generally produce unsatisfactory results since the whole contents tend to be quite different from the initial random values (which will be mentioned in Section 4 and Figure 8) and it may cause the rim effect to become distinct. According to our experience, more suitable approach is to choose just the left border and the top border of the training patch as the training data, as shown in Figure 5. An additional reason for selecting just the border part is that the image scanning convention adopted in this study is from top-left to bottom-right, as shown in Figure 6.

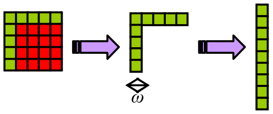


Fig. 5. Using only the left border and the top border for each search patch, which has a thickness of ω pixels (ω=2 pixels in this study) and is called a Γ-shaped pattern. Each pattern contains *K* pixels, where $K = \omega \times (Wp + Hp - \omega)$. So the size of each patch is reduced from a *Wp*x*Hp* dimensional (pixel) vector to a *K*-dimensional (pixel) vector ($P_1 \ldots P_K$). ($K << Wp$xHp).

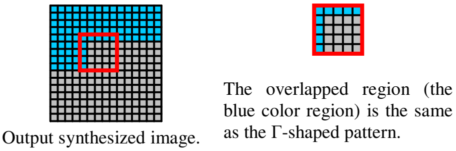

Output synthesized image.
The overlapped region (the blue color region) is the same as the Γ-shaped pattern.

Fig. 6. For the output synthetic image and the search window (or patch), the blue color region has been already completed; the gray color region has a random value originally, and is not synthesized yet; and the red color window indicates the region undergoing synthesis

3.3 Principal Component Analysis (PCA)

PCA is applied to the training data to obtain their eigenspace, Ψ, as shown in Figure 7. Two important properties of PCA are employed to obtain the best performance: (1) reducing the dimensions of the data representation from K dimensions to N dimensions, where $N<K$, as shown in Figure 7. This can reduce most of the time complexity operations, and hence dramatically increases the performance; (2) recombining the appearance of the features while maintaining the coherence of the characteristic content. After the PCA process and a sort based on the eigenvalues with corresponding eigenvectors, it is found that the first several eigenvectors control the global geometrical structure, while the middle eigenvectors control the local features. Meanwhile, some noises are controlled by the last few eigenvectors. These noises cause the photograph to appear truer, but have no influence on the geometric structure. Therefore a good matching structure need only compare the first several eigenvectors. Consequently, this study uses only the first N eigenvectors, whose corresponding eigenvalues occupy 98% of total eigenvalues, for comparison purposes to identify the patch which results in the best matching of the geometrical structure. This approach makes the result more fitted visually.

$$\begin{bmatrix} P_{11} & P_{12} & \cdots & \cdots & \cdots & P_{1M} \\ P_{21} & P_{22} & \ddots & & & P_{2M} \\ \vdots & & & \ddots & & \vdots \\ P_{K1} & P_{K2} & \cdots & \cdots & \cdots & P_{KM} \end{bmatrix} \xrightarrow{\textbf{PCA}} \Psi = \begin{bmatrix} E_{11} & E_{12} & \cdots & E_{1K} \\ E_{21} & E_{22} & \cdots & E_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ E_{N1} & E_{N2} & \cdots & E_{NK} \end{bmatrix}$$

Fig. 7. *M* patches can be obtained from Section 3.1. Each patch has *K* elements (see Section 3.2.) During training, PCA is used to transform the original $K \times M$ matrix to an $N \times K$ matrix, where $N<K<<M$. And there are *N* eigenvectors.

Because there are M patches for any single input image, and K elements in each patch P ($P=[P_1 \ldots P_k]^T$) projected onto N eigenvectors ($E_{1i} \ldots E_{Ni}$, where $i=1\sim K$), as shown in Equ. (1).

$$\begin{bmatrix} E_{11} & E_{12} & \cdots & \cdots & \cdots & E_{1K} \\ E_{21} & E_{22} & \ddots & & & E_{2K} \\ \vdots & \vdots & & \ddots & & \vdots \\ E_{N1} & E_{N2} & \cdots & \cdots & \cdots & E_{NK} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ \vdots \\ \vdots \\ P_K \end{bmatrix} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_N \end{bmatrix} \tag{1}$$

So the corresponding N-dimensional weight vector can be obtained. This weight vector represents the information relating to the origin of each patch. Therefore, there are a total of M N-dimensional weight vectors.

3.4 Vector Quantization (VQ)

Searching for the best matching pattern from the eigenspace Ψ during the synthesis process is a computationally expensive task. In order to speed up the synthesis process, the training data is initially projected onto the eigenspace Ψ in order to retrieve the weight vectors. The data is then separated into C clusters by means of VQ. Using this approach, the computational time can be reduced from $O(M)$ to $O(\sqrt{M})$.

4 Texture Synthesis Module

The texture synthesis module comprises four steps, as shown in Figure 8. Given a predefined size (*Wo*x*Ho* pixels) of the RGB image, the RGB values of each pixel in the output image are initialized randomly from 0~255, individually. The whole content of the output image is retrieved by shifting a search window (or patch) of size *Wp*x*Hp* over this image buffer in the scan-line order mentioned previously and acquiring the Γ-shaped pattern from the region of the search window. Traditionally, synthesis processes using PCA process reconstructed the desired patches via the linear combination of the first N eigenvectors. This study has attempted to project the whole patch having random value for each pixel onto the training patch's eigenspace, and then to reconstruct the whole patch in one step for the output synthetic image. However, in implementation, it was found that this approach produced a blurred result and that much of the original detail was lost.

Accordingly, an indirect approach was developed in which the Γ-shaped pattern of each search window, which contains random value for each pixel, was first projected onto the eigenspace Ψ to have the corresponding weight vector. Second, based on the similarity measure of using the sum of squared difference (SSD) between the weight vector of this search window and M patches in the original input image, this weight vector was classified to the closest cluster and then this cluster was searched to find the best matching weight vector. Third, the matching patch, which corresponds to the best matching weight vector, in the original input image was taken to fill the region of this search window.

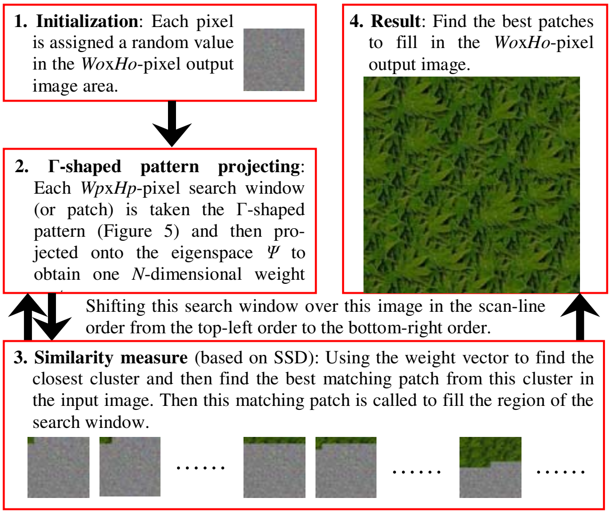

1. Initialization: Each pixel is assigned a random value in the WoxHo-pixel output image area.
4. Result: Find the best patches to fill in the WoxHo-pixel output image.
2. Γ-shaped pattern projecting: Each WpxHp-pixel search window (or patch) is taken the Γ-shaped pattern (Figure 5) and then projected onto the eigenspace Ψ to obtain one N-dimensional weight
Shifting this search window over this image in the scan-line order from the top-left order to the bottom-right order.
3. Similarity measure (based on SSD): Using the weight vector to find the closest cluster and then find the best matching patch from this cluster in the input image. Then this matching patch is called to fill the region of the search window.
……
……
……

Fig. 8. Framework of the texture synthesis module. All images here have the same size of *Wo*x*Ho* pixels.

5 Image Replacement Through Texture Synthesis

Both texture analysis module and texture synthesis module can be applied to the image replacement of the particular region for a given image, as shown in Figure 1. Here an original input image is selected and is annotated as I_0 (Figure 1.a). And the regions containing the replaced pixels are manually acquired and are called inverse matte, β_0 (Figure 1.b).

This matte is binary, such that the white regions which are to be retained are set to 1, and are the known regions, while the black regions which are to be replaced regions are set to 0, and are the replaced region. In addition, the replaced regions can comprise many subregions, must contain the removable objects, can exceed the boundaries of the removable objects, and can be of any shapes. But too many or too large replaced regions will lead to a poor quality result. The known region serves as the source of the replaced regions. The analysis module developed in Section 3 is used for training, while the synthesis module presented in Section 4 is employed for filling the replaced region. But some revisions are performed to be suitable for the process of image replacement.

5.1 Preprocessing Using Multi-resolution Approach

The multi-resolution approach makes the input image I_0 and the inverse matte β_0 to do l times down-sampling ↓, then to get each level of input image I_i and inverse matte β_i, i=0~l, as shown in Figure 9. The value of the level i can serve as the initial values of the level i-1. And the size of the replaced region at the lowest level l of image I_l is less than that of one patch, so we need not assign an initial random value or approximate value to each pixel in the replaced region. Instead, relevant definition values located at the background region are used to be the initial values of the search patch. In addition, B_i denotes the background region, which must be preserved for training data, as in Equation (2),

$$B_i = I_i \beta_i \quad , \quad i = 0 \sim l \tag{2}$$

and the foreground region F_i, as in Equation (3),

$$F_i = I_i \overline{\beta_i} \quad , \quad i = 0 \sim l \tag{3}$$

is then utilized to search the background B_i to locate the best patch with which to fill in the replaced region.

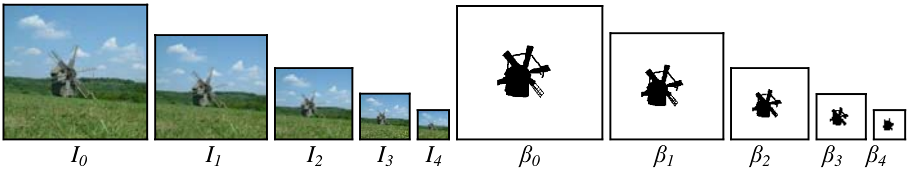


Fig. 9. We have input image I_0 and inverse matte β_0, and then do l times down-sampling ↓ to get each level of input image I_i and inverse matte β_i, i=0~l

5.2 Training Process Based on Background Region

For the training process, we get the whole piece of the patches from the background regions, which are preserved for training data from the level 1 image I_1 to the level l image I_l. We drop the patches at the level 0 image I_0 for the following reasons: (1) The total amount of patches are too many (usually 100~300 thousands of patches) and cost much time in many unnecessary operators. (2) Using plenty of patches is unnecessary to fill in the removed (or replaced) region. VQ needs to separate such many patches into more clusters for speeding up searching the matching patches. Furthermore, more clusters raise the probability of matching the wrong patches, and the time of VQ training increases exponentially following the increasing cluster. (3) The level 0 image I_0 has more noise influence on the process of PCA training. That results easily in getting the wrong information. (4) While reconstructing the lower level images (ex. I_2~I_l), the system needs stronger structural patches to fill the replaced region. Because of applying multi-resolution to input image I_0, the patches at the lower level of image structurally are more powerful.

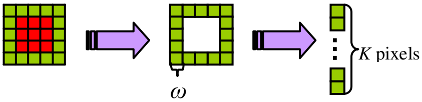


Fig. 10. Acquire the four borders with thickness ω (ω=2) pixels for each search patch. There are K pixels in each patch, where $K = 2\times\omega\times(Wp+Hp-2)$.

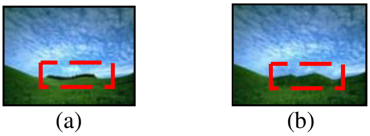


Fig. 11. (a) Input image with the removed object between two different appearances (textures) and including light effect. (b) Output reconstructed image using the modified patch shown in Figure 10.

The training process of image replacement is the same as that of the texture analysis in Section 3. However, if the same Γ-shaped pattern of the search window (or patch) is applied (Figure 5), the fragment between two different textures will be mapped by only one texture rather than being mapped by the mixing of two different textures. Hence, the search window is modified, as shown in Figure 10, and the performance looks promising as shown in Figures 11, 13, and 14. Here, we analyze the problem when applying the Γ-shaped pattern to the process of image replacement. For example, in Figure 11(a), the top half of the image is sky, the bottom half of the image is grass, and the middle part shows some trees. If our previous proposed approach is used to remove the tree region and to fill in the removed region by using the Γ-shaped pattern to reconstruct the image. It can be seen that the resulting image is filled well near the sky in the original location of the trees, but the region of the image toward the grass is filled in with sky data and hence does not merge satisfactorily with the grass region.

In an alternative approach, two patterns are used, i.e. the top-left Γ-shaped pattern and the bottom-right Γ-shaped pattern. Initially, the top-left Γ-shaped pattern is used to reconstruct the image in a top-down order. The bottom-right Γ-shaped pattern is then employed to reconstruct the image in a bottom-up order. The two steps are then repeated. In the case of Figure 11, the replaced region is initially filled completely with sky data and then filled in completely with grass data. As the two-step process is repeated, these two actions are repeated continuously and convergence cannot be achieved.

5.3 Completion Process

In order to practice for convenience, one minimum rectangular boundary covering the entire replaced region is identified. Then the same texture synthesis process as in Section 4 is applied to this rectangular region. We only change the compared part to four borders of the patch, whose thickness is ω pixels, as shown in Figure 10. In our algorithm, the reconstructed order is from the level l image I_l to the level 0 image I_0. First, the system reconstructs the level i (i=l) image I_i. The replaced part is reconstructed (or synthesized), called $I_i^{'}$. Only the reconstructed part of $I_i^{'}$ is preserved at position F_i. The other parts are replaced by the original background B_i. This gives the first step of the reconstructed image C_i, as in Equation (4).

$$C_i = I_i \beta_i + I'_i \overline{\beta_i} \tag{4}$$

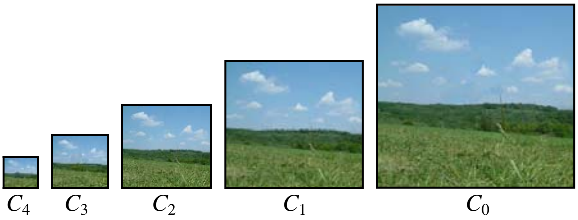


Fig. 12. Show reconstructed image C_i from the lowest level 4 to the origin level 0, i=0~4. C_i does up-sampling ↑, and then serves I_{i-1} as initial value for searching the matching patch to fill the removed (or replaced) region of C_{i-1}.

The system applies the up-sampling ↑ technique from the reconstructed image C_i at level i and proceeds to only replace the level i-1 image $I_{i\text{-}1}$ at position $F_{i\text{-}1}$, as in Equation (5).

$$I_{i-1} = I_{i-1}\beta_{i-1} + (C_i \uparrow)\overline{\beta}_{i-1} \tag{5}$$

Then the system repeats above process of Equations (4) and (5) from the level l-1 image I_{l-1} (i=l-1) to the level 0 image I_0 (i=0). But the process of Equation (5) does not require at the last time (i=0). According to our experiments, when the system reconstructs directly the level 0 image I_0, the output image is often converge to the worse result or needs to repeat many times (usually 70~100 times) of reconstructing processes. But using the multi-resolution approach, our system can reconstruct image not only fast but also promising, as shown in Figure 12.

6 Experimental Results

In acquiring the experimental statistics presented below, the process was performed ten times and the average time calculated. The experiments were performed on a personal computer with a AMD K8 3200+ (2.0 GHz) processor, 512MB DDR SDRAM, and the Windows XP SP2 operating system. The current system was written in Visual C++.

Figure 13 shows various results by using our developed modules in Section 3 and Section 4. The width of the patch, *Wp*, is 32 pixels, the height of the patch, *Hp*, is 32 pixels, and the thickness of the patch, ω, is 2 pixels. As shown in Table 1, the training process can be completed rapidly, and the time complexity is proportional to the size of the input image. Furthermore, the proposed method can synthesize a seamless texture rapidly, as shown in Table 2. Table 3 and Table 4 show the processing information in Figure 9 and Figure 12. The required time is also proportioned to the size of the output image and the value of *N* for the dimension of eigenspace. Figure 18 shows various results of Section 5. The size of the patches is modified such that *Wp* is 16 pixels and *Hp* is 16 pixels. However, the same thickness as prescribed in Section 5 is retained. The results have a higher quality and are also completely rapidly.

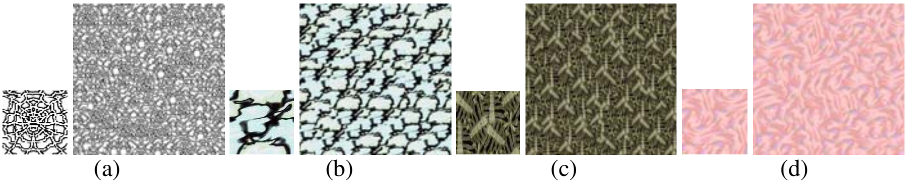


Fig. 13. Left column: Input images (a)~(c) (size: 64x64 pixels) and (d) (size: 128x128 pixels). Right column: Output images using the proposed approach described in Sections 3 and 4 (size: 300x300 pixels).

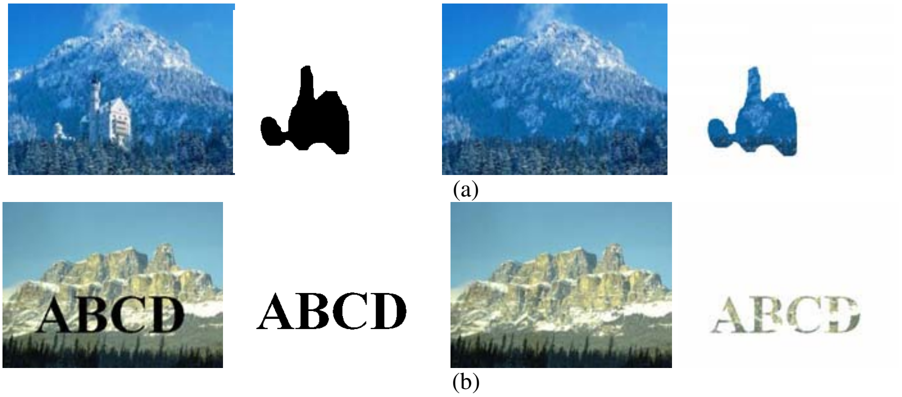


Fig. 14. From left to right columns: input images, inverse mattes, output images, and the synthesis results used to replace the replaced regions

Table 1. Average training time in Section 3 for input images of various sizes. Units of time: milliseconds (ms).

Size of input image (pixels)	64x64	96x96	128x128
PCA	250	437	875
Projecting to eigenspace	359	703	1656
VQ	203	391	953
Total time (ms)	**812**	**1531**	**3484**

Table 3. The synthesized information of each level in Figure 9 and Figure 12 for multi-resolution approach. "~0" means close to 0 ms.

Level	Width (pixels)	Height (pixels)	Number of whole patches	Synthesis time (ms)
0	392	364	114034	63
1	196	182	24272	16
2	96	91	4024	~0
3	49	46	110	~0
4	25	23	0	~0

Table 2. Average synthesizing time in Section 4 for output images of various sizes

Size of output image (pixels)	200x200	300x300	400x400	600x600
Synthesis time (ms)	11	37	59	140

Table 4. The time in Figure 9 and Figure 12 for various processes. The total time of the training and synthesis processes include the processes of the multi-resolution approach, PCA, projection (the patches are projected onto eigenspace Ψ to obtain the weight vectors), VQ, and synthesis.

Method		Time (ms)	
Data type		**Gray value**	**RGB space**
Training process	Multi-resolution	32	31
	PCA	891	7719
	Projecting to eigenspace	1828	4906
	VQ	421	375
Synthesis process	Synthesis	78	108
Total time		**3250**	**13139**

7 Conclusions

We developed a system including two modules: the texture analysis and synthesis modules. This system is able to be applied to the two different purposes: the synthesis of a large image, and the replacement of local removed region. According to the training non-periodic or periodic pattern, we use different sampling methods to obtain different amount of patches in order to reduce the emergences of the seams of the output synthesized image. And because the analysis module can reduce dimensions of the training data and cluster these data, so the synthesis module can synthesize a large output image very fast and keep geometrical structures and veins continuous. The same process can also be used to replace the removed regions. Here, the multi-resolution approach is applied to the image replacement without needing to assign initial random values or approximate values. The down-sampling step is used for the analysis process as compiling the training data, and the up-sampling step is used for the reconstructing process as assigning initial values. So this approach enables the system to handle the large removed region and obtain more realistic image (or textures) quickly.

References

1. Ashikhmin, M., "Synthesizing Natural Textures," In ACM Symposium on Interactive 3D Graphics, pp. 217–226, 2001.
2. Drori, I., Cohen-Or, D., and Yeshurun, H., "Fragment-Based Image Completion," ACM Trans. on Graphics (SIGGRAPH), Vol. 22, No. 3, pp. 303-312, 2003.

3. De Bonet, J. S., “Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images,” In ACM SIGGRAPH, Computer Graphics Proceedings, pp. 361–368, 1997.
4. Efros, A. A., and Freeman, W. T., “Image Quilting for Texture Synthesis and Transfer,” In ACM SIGGRAPH, Computer Graphics Proceedings, pp. 341–346, 2001.
5. Efros, A. A., and Leung, T. K., “Texture Synthesis by Non-parametric Sampling,” In International Conference on Computer Vision, pp. 1033–1038, 1999.
6. Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F., “The Lumigraph,” In Proceedings of ACM SIGGRAPH 96, ACM Press, pp. 43–54, 1996.
7. Igehy, H., and Pereira, L. “Image Replacement through Texture Synthesis,” In IEEE International conference on Image Processing, Vol. 3, pp. 186–189, 1997.
8. Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A., “Graphcut Textures: Image and Video Synthesis Using Graph Cuts,” ACM Trans. on Graphics (SIGGRAPH), Vol. 22, No. 3, pp. 277–286, 2003.
9. Liang, L., Liu, C., Xu, Y., Guo, B., and Shum, H.-Y., “Real-Time Texture Synthesis Using Patch-Based Sampling,” ACM Trans. on Graphics, Vol. 20, No. 3, pp. 127-150, 2001.
10. Liu, Y., Lin, W.-C., and Hays, J., “Near-Regular Texture Analysis and Manipulation,” ACM Trans. on Graphics (SIGGRAPH), Vol. 23, No. 3, pp. 368–376, 2004.
11. Nealen, A., and Alexa, M., “Fast and High Quality Overlap Repair for Patch-Based Texture Synthesis,” In Proceedings of Computer Graphics International, pp. 582-585, 2004.
12. Wei, L.-Y. “Texture Synthesis from Multiple Sources,” ACM Trans. on Graphics (SIGGRAPH), 1-1, 2003.
13. Wei, L.-Y., and Levoy, M., “Fast Texture Synthesis using Tree-structured Vector Quantization,” In ACM SIGGRAPH, Computer Graphics Proceedings, pp. 479–488, 2000.
14. Wu, Q., and Yu, Y., “Feature Matching and Deformation for Texture Synthesis,” ACM Trans. on Graphics (SIGGRAPH), Vol. 23, No. 3, pp. 362-365, 2004.
15. Yamauchi, H., Haber, J., and Seidel, H.-P., “Image Restoration using Multiresolution Texture Synthesis and Image Inpainting,” In Proceedings of Computer Graphics International, pp. 120-125, 2003.
16. Zelinka, S., and Garland, M., “Towards Real-Time Texture Synthesis with the Jump Map,” Eurographics Organization, pp. 99-104, 2002.

Histogram Features-Based Fisher Linear Discriminant for Face Detection

Haijing Wang[1], Peihua Li[2], and Tianwen Zhang[1]

[1] Harbin Institute of Technology, School of Computer Science and Technology,
P.O.Box 1071, Harbin, Heilongjiang 150001, China
ninhaijing@yahoo.com

[2] Heilongjiang University, College of Computer Science and Technology, China
peihualj@hotmail.com

Abstract. The face pattern is described by pairs of template-based histogram and Fisher projection orientation under the paradigm of AdaBoost learning in this paper. We assume that a set of templates are available first. To avoid making strong assumptions about distributional structure while still retaining good properties for estimation, the classical statistical model, histogram, is used to summarize the response of each template. By introducing a novel "integral histogram image", we can compute histograms rapidly. Then we turn to Fisher linear discriminant for each template to project histograms from $d-$dimensional to one-dimensional subspace. Best features, used to describe face pattern, are selected by AdaBoost learning. The results of preliminary experiments demonstrate that the selected features are much more powerful to represent the face pattern than the simple rectangle features used by Viola and Jones and some variants.

1 Introduction

Face detection is one of the visual tasks which humans can do effortlessly. Yet in computer vision community, this task is not easy. As a visual frontend processor, a face detection system should be able to achieve the task regardless of illumination changes, and orientation, position, scale, expression variations of human faces.

Viola and Jones [1] present the first highly accurate as well as real-time frontal face detector at 15 frames per second for 384 by 288 image. They use simple rectangle features to describe face pattern that can be computed rapidly via "integral image". Best features are selected automatically with AdaBoost algorithm, and cascade architecture is adopted to speed up detection. Many researchers present their works following the idea of Viola and Jones, mainly addressing two issues: (i) how to develop more powerful features to represent face pattern, and (ii) how to classify examples based on the chosen representation.

From the view of feature selection, Murphy *et al.* [2] use a set of filters to convolve the image, and utilize the second and the fourth moments to calculate features from the different patches on the filtered images. Levi and Weiss [3]

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 521–530, 2006.

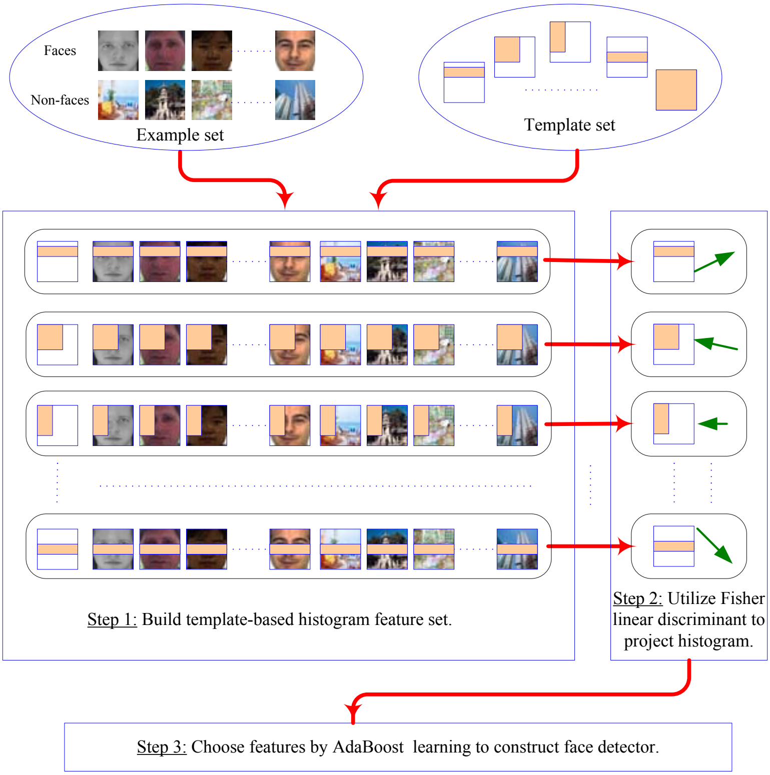

Faces
Non-faces
Example set
Template set
Step 1: Build template-based histogram feature set.
Step 2: Utilize Fisher linear discriminant to project histogram.
Step 3: Choose features by AdaBoost learning to construct face detector.

Fig. 1. Framework of face detection algorithm

take local edge orientation histograms (EOH) as features. Wang *et al.* [4] take histograms computed from a set of rectangles in the filtered images as features. For the second issue, Wu *et al.* [5] propose a cascade learning algorithm based on forward feature selection which is two orders of magnitude faster than the Viola-Jones' approach and yields classifiers with similar quality. Li *et al.* [6] present the first real-time multiview face detection system by FloatBoost. Torralba *et al.* [7] propose a multi-class boosting procedure (joint boosting) that reduces both the computational and sample complexity, by finding common features that can be shared across the classes.

In this paper, we propose the novel feature, *template-based histogram along with Fisher projection orientation*, for face detection in the paradigm of AdaBoost algorithm. The results of preliminary experiments demonstrate that the selected features are much more powerful to represent the face pattern than the simple rectangle features used by Viola and Jones and some variants.

Our face detection algorithm consists of three major steps (see Fig. 1), as listed below.

(i) Build template-based histogram feature set. We assume that a set of templates are available first, then summarize the response of each template patch using one histogram, which represents marginal distribution of the patch. To speed up histogram computation, we extend "integral image" proposed by Viola and Jones [1] from one-dimensional to $n-$dimensional integral image, called "integral histogram image".

(ii) Utilize Fisher linear discriminant to project histogram. Fisher linear function yields the maximum ratio of between-class scatter to within-class scatter. Thus, for each template patch, we turn to Fisher linear discriminant to find a projection orientation of histograms. Two classes (faces and non-faces) are well separated by this Fisher projection orientation.

(iii) Choose features by AdaBoost. The best features to separate face and non-face examples are chosen by AdaBoost learning.

The paper is structured as follows. In Section 2, we present the template-based histogram feature set. Fisher linear discriminant is used to project histogram in Section 3. The AdaBoost training to choose best features is described in Section 4. Experimental results are shown in Section 5. Finally, conclusions and directions for future research are given.

2 Template-Based Histogram Feature Set

We assume that a set of reference patterns (templates) are available in this section. To seek statistical models that avoid making strong assumptions about distributional structure while still retaining good properties for estimation, the best compromise we found was histograms. To speed up histogram statistics, we extend "integral image" proposed by Viola and Jones from one-dimensional to $n-$dimensional integral image, called "integral histogram image".

Taking a 64×64 image for example, there are totally 892 different rectangle templates. Fig. 2 shows 59 reference patterns with the top left point $(0, 0)$. The orange rectangles are the masks used to calculate histogram features. Other rectangle templates are created in a step of eight pixels. Each template includes 256 pixels at least. Both width and height of each template are no less than eight pixels.

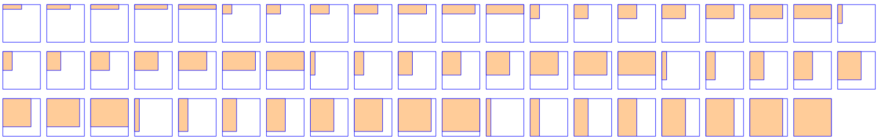


Fig. 2. Example templates with the top left point $(0, 0)$ for 64×64 image. The orange rectangles are the masks used to calculate histogram feature.

Our histogram statistics can be computed rapidly using an intermediate representation for the image which is called the "integral histogram image" [4] (see Fig. 3). Given a $p \times q$ image, the integral histogram image I is $(p + 1) \times (q + 1)$ arrays of length d (dimension of histogram). The integral histogram image $I_{x,y}[k]$

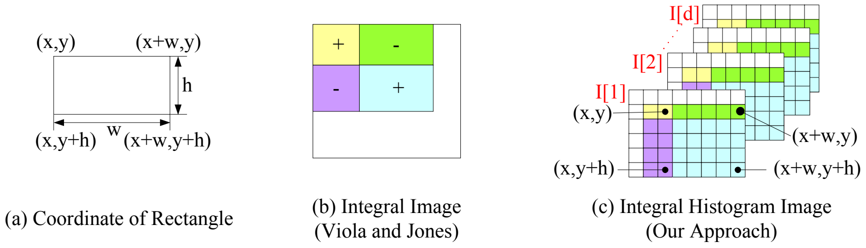

(x,y)
(x+w,y)
h
w
(x,y+h)
(x+w,y+h)
(a) Coordinate of Rectangle
+
-
-
+
(b) Integral Image (Viola and Jones)
I[d]
I[2]
I[1]
(x,y)
(x+w,y)
(x,y+h)
(x+w,y+h)
(c) Integral Histogram Image (Our Approach)

Fig. 3. Integral Image vs. Integral Histogram Image. Based on the same rectangle region shown in (a), (b) gives the integral image proposed by Viola and Jones [1]; and our integral histogram image is shown in (c).

at location (x, y) corresponds to the histogram of the image above and to the left of (x, y), inclusive:

$$I_{x,y}[k] = \sum_{x' \leq x, y' \leq y} \delta(x', y'), \; k = 1, \ldots, d \tag{1}$$

where $\delta(x', y') = 1$ if the intensity of pixel at location (x', y') belongs to the k-th bin of histogram; otherwise let $\delta(x', y') = 0$. Using the following pair of recurrences:

$$\begin{aligned} i_{x,y}[k] &= i_{x,y-1}[k] + \delta(x, y) \\ I_{x,y}[k] &= I_{x-1,y}[k] + i_{x,y}[k], k = 1, \ldots, d \end{aligned} \tag{2}$$

where $i_{x,0}[k] = 0$ for any x and k, the integral histogram image can be computed in one pass over the original image. The histogram $h_r[k](k = 1, \ldots, d)$ of any rectangle region r can be determined in $(4 \times d)$ array references (see Fig.3 and Equ.(3)) by integral histogram image for $k = 1, \ldots, d$:

$$h_r[k] = I_{x+w,y+h}[k] - I_{x+w,y}[k] - I_{x,y+h}[k] + I_{x,y}[k] \tag{3}$$

where $I_{x,0}[k] = I_{0,y}[k] = 0$, w and h are the width and height of rectangle r, respectively.

3 Histogram Projection by Fisher Linear Discriminant

Different from PCA (principal component analysis), which seeks directions that are efficient for representation, Fisher linear discriminant seeks directions that are efficient for discrimination. Its linear function yields the maximum ratio of between-class scatter to within-class scatter. Thus, we turn to Fisher linear discriminant for each template to find an projection orientation of histograms by which two classes (faces and non-faces) are well separated. That is, each template is corresponding to one Fisher projection orientation. Moreover, the classification task can been converted from a $d-$dimensional problem to a one-dimensional one.

We consider our problem as projecting template-based histograms from d-dimensional subspace onto a line for subsequent AdaBoost learning. For any one template, suppose that we have a set of n $(n_1 + n_2 = n)$ $d-$dimensional histograms $\mathbf{h}_1, \ldots, \mathbf{h}_n$, where n_1 is the size of the subset $\mathcal{H}_1$ labeled τ_1 (face class) and n_2 the subset $\mathcal{H}_2$ labeled τ_2 (non-face class). If we form a linear combination of the components of $\mathbf{h}_i$, we obtain the scalar dot product

$$z_i = \mathbf{v}^t \mathbf{h}_i \tag{4}$$

and a corresponding set of n projected points $z_1, \ldots, z_n$ divided into the subsets $\mathcal{Z}_1$ and $\mathcal{Z}_2$. Geometrically, if $||\mathbf{v}|| = 1$, each z_i is the projection of the corresponding $\mathbf{h}_i$ onto a line in the direction of $\mathbf{v}$.

The Fisher linear discriminant employs the linear function Equ.(4) for which the criterion function

$$J(\mathbf{v}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \tag{5}$$

is maximized. That is, the $\mathbf{v}$ maximizing $J(\cdot)$ leads to the best separation between the two projected sets ($\mathcal{H}_1$ and $\mathcal{H}_2$). Here $\tilde{m}_i$ is the mean for the projected histograms ($\mathcal{Z}_i$) of set $\mathcal{H}_i$. We define the scatter for projected histograms labeled τ_i by

$$\tilde{s}_i^2 = \sum_{z \in \mathcal{Z}_i} (z - \tilde{m}_i)^2, i = 1, 2 \tag{6}$$

Thus, $(1/n)(\tilde{s}_1^2 + \tilde{s}_2^2)$ is an estimate of the variance of all histograms, and $\tilde{s}_1^2 + \tilde{s}_2^2$ is called the total within-class scatter of the projected samples.

According to the generalized Rayleigh quotient known in mathematical physics, the criterion function $J(\cdot)$ shown in Equ.(5) can be written as

$$J(\mathbf{v}) = \frac{\mathbf{v}^t \mathbf{S}_B \mathbf{v}}{\mathbf{v}^t \mathbf{S}_V \mathbf{v}} \tag{7}$$

$\mathbf{S}_V$ is called the within-class scatter matrix defined by

$$\mathbf{S}_V = \mathbf{S}_1 + \mathbf{S}_2, \quad \mathbf{S}_i = \sum_{\mathbf{h} \in \mathcal{H}_i} (\mathbf{h} - \mathbf{m}_i)(\mathbf{h} - \mathbf{m}_i)^t \tag{8}$$

where $\mathbf{m}_i$ is the $d-$dimensional histogram mean of set $\mathcal{H}_i$. $\mathbf{S}_B$ is called the between-class scatter matrix defined by

$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t \tag{9}$$

Now we get the solution for the $\mathbf{v}$ that optimizes $J(\cdot)$ as:

$$\mathbf{v} = \mathbf{S}_V^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \tag{10}$$

which is sometimes called the canonical variance. Thus the classification has been converted from a $d-$dimensional problem to a hopefully more manageable one-dimensional one by Equ.(4).

4 Feature Selection by Gentle AdaBoost Algorithm

Boosting algorithm, proposed in the Computational Learning Theory literature [8], is a method to find a highly accurate hypothesis by combining many "weak" hypotheses, each of which is based on the reweighted version of the training data, and only moderately accurate. The adaptive version of Boosting is called AdaBoost [9]. We choose template-based Fisher projection orientation and corresponding threshold value to construct the weak hypothesis, which are used to separate face and non-face examples, by each round of AdaBoost learning. We and others [10], have found that Gentle AdaBoost gives higher performance than Discrete AdaBoost and Real AdaBoost, and requires fewer iterations to train. We will briefly present the Gentle AdaBoost below.

Given a set of training examples $\mathcal{X}$ with its weight distribution D, the Boosting procedure computes a weak hypothesis $f : \mathcal{X} \mapsto R$, where the sign of f is the predicted label $\lambda \in \{\tau_1, \tau_2\}$ of the example $x \in \mathcal{X}$, and the magnitude $|f(x)|$ is the confidence in this prediction. This is called Real AdaBoost (RAB) [11]. The simplest case, $f : \mathcal{X} \mapsto \{-1, +1\}$, is called Discrete AdaBoost (DAB) [9]. Let $f_1, f_2, \ldots, f_T$ stand for a set of learned weak hypotheses, thus the ensemble hypothesis is $F(x) = \mathcal{E}[\lambda|x] = \sum_{t=1}^{T} f_t(x)$, where $\mathcal{E}$ represents the expectation. Suppose we have a current estimation F and seek an improved estimation $F + f$ by minimizing criterion shown in Equ.(11):

$$J(F + f) = \mathcal{E}[e^{-\lambda(F(x)+f(x))}] \tag{11}$$

RAB optimizes J with respect to $f(x)$ at each iteration. Gentle AdaBoost (GAB) [12], a modified version of RAB, takes adaptive Newton steps to minimize $J(F + f)$ by

$$F(x) \leftarrow F(x) + \frac{\mathcal{E}[e^{-\lambda F(x)}\lambda]}{\mathcal{E}[e^{-\lambda F(x)}]} = F(x) + \mathcal{E}_\omega[\lambda|x] \tag{12}$$

Here the notation $\mathcal{E}_\omega[\lambda|x]$ refers to a weighted conditional expectation, and the weight is updated by Equ.(13).

$$\omega \leftarrow \omega \cdot e^{-\lambda f(x)} \tag{13}$$

Therefore, the weak hypothesis $f(x)$ is written as

$$f(x) = \mathcal{E}_\omega[\lambda|x] = \frac{\mathcal{E}[e^{-\lambda F(x)}\lambda]}{\mathcal{E}[e^{-\lambda F(x)}]} \tag{14}$$

To get optimized $f(x)$, we expand $J(F + f)$ to the second order about $f(x) = 0$. Minimizing pointwise with respect to $f(x)$, there is

$$\hat{f}(x) = \arg\min_f \mathcal{E}_\omega[(\lambda - f(x))^2|x] \tag{15}$$

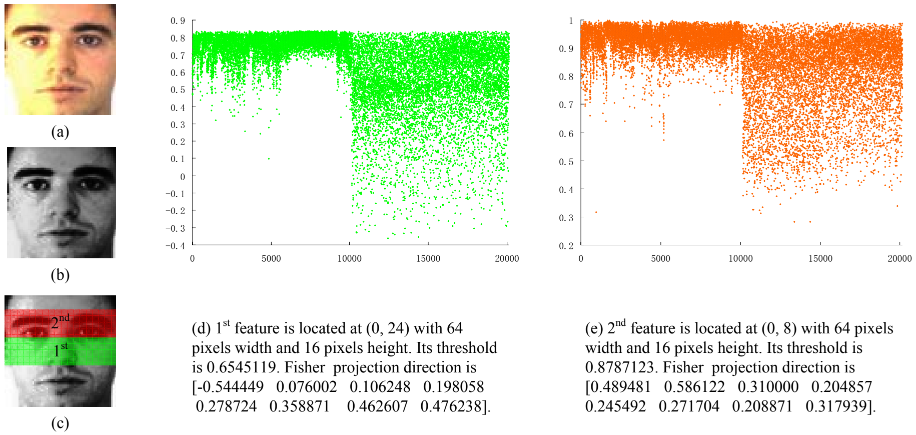
Equ.(15) shows the way to obtain the weak hypothesis $f(x)$.

We utilize Gentle AdaBoost (GAB) to train the final cascade face detector [13]. Each trained classifier $f(x)$ produces a weak classification rule based on Equ.(15) with one histogram Fisher projection orientation described in the previous section. The weight distribution of examples is updated via Equ.(13)

at each round of GAB learning. The threshold value of the final strong classifier is decided by the prescribed hit ratio of the strong classifier $F(x)$ to the training example set $\mathcal{X}$. The construction of the final cascade detector depends on the ratio of false positives for the training set.

5 Experimental Results

In this section, we first introduce the training data set and feature set. Then learning results and detection results are described.



(d) 1st feature is located at (0, 24) with 64 pixels width and 16 pixels height. Its threshold is 0.6545119. Fisher projection direction is [-0.544449 0.076002 0.106248 0.198058 0.278724 0.358871 0.462607 0.476238].

(e) 2nd feature is located at (0, 8) with 64 pixels width and 16 pixels height. Its threshold is 0.8787123. Fisher projection direction is [0.489481 0.586122 0.310000 0.204857 0.245492 0.271704 0.208871 0.317939].

Fig. 4. (a) and (b) are the original positive image and the image after histogram equation, respectively. (c) shows the locations of the first and second features. The projections of all training samples corresponding to the first and second features of the detector are shown in (d) and (e). X axis represents the sample ID. The first 10, 135 samples are positives (faces) and the Id from 10, 136 to 20, 135 represents negatives (non-faces). Y axis is the Fisher linear projection value.

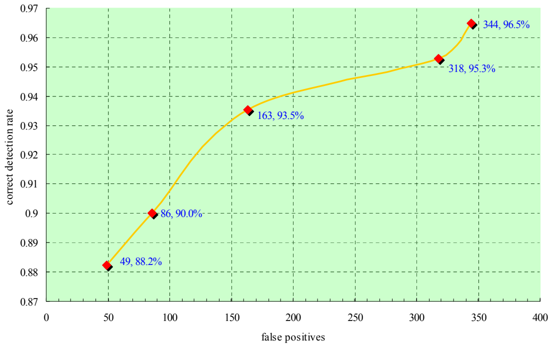


Fig. 5. ROC curves for our face detector on the CMU new test set. X axis and Y axis represent detection rate and false positives, respectively. The detection rate achieves 90% with 86 false detections.

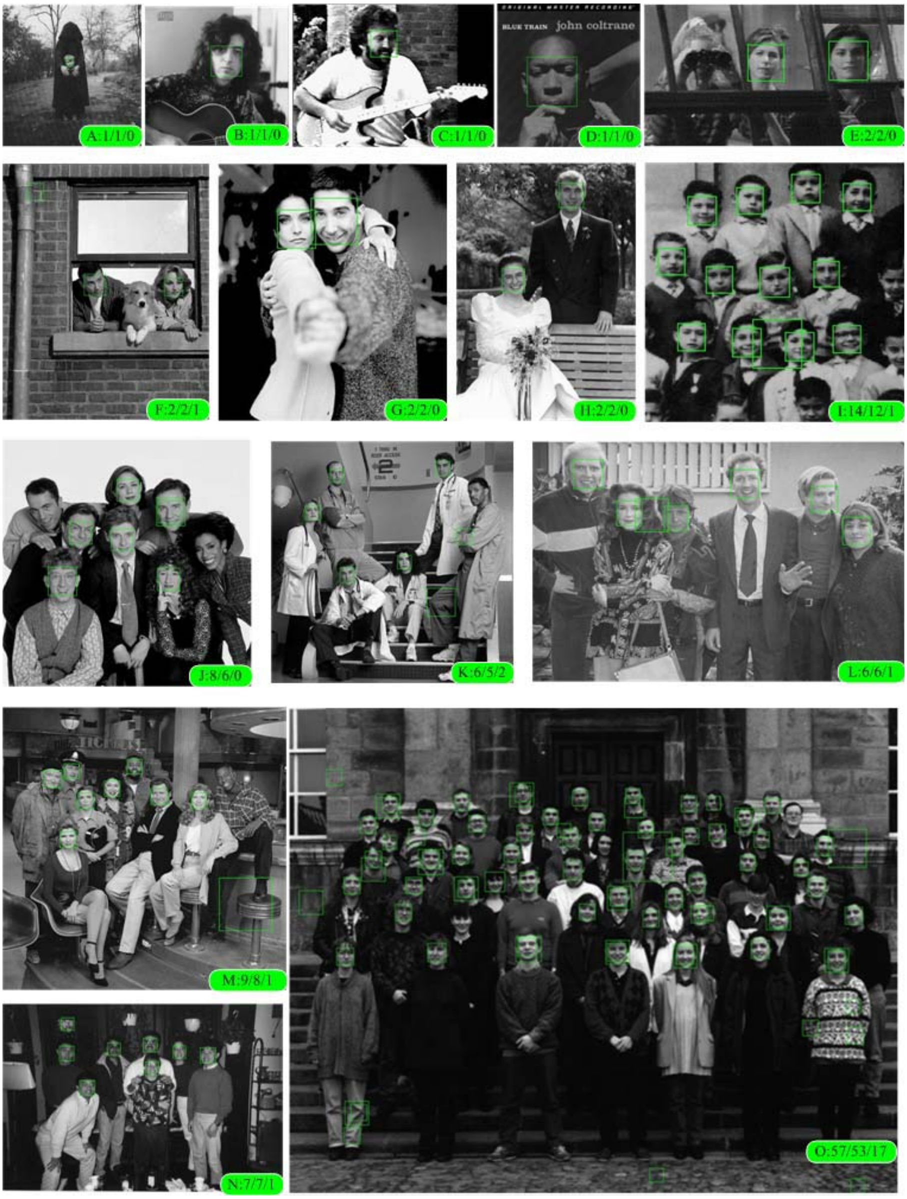


Fig. 6. Output of our face detector on a number of test images from the CMU new test set

We crop 10, 135 frontal face images as training samples. The negative samples are collected by selecting random sub-windows from a set of 24, 621 images which do not contain faces. For each layer training, the maximum size of the negative set is 10, 000. Each sample is scaled to 64 by 64 pixels, which includes enough rich information for template-based histogram calculation. We take histogram equalization for both training samples and test samples to make each image with equally distributed brightness levels over the whole brightness scale. One example image (see Fig. 4(a)), preprocessed by histogram equation, is given in Fig. 4(b).

Given the base resolution of the detector is 64 × 64, our feature set only includes 892 template-based features, which is far less than 45,396, the size of Viola and Jones' 24 × 24 detector. We calculate eight dimensional histogram at each template location.

Our cascade detector only includes 17 layers with 2347 features. The first and second features are shown in Fig. 4. However, the final detector of Viola and Jones is a 38 layer cascade of classifiers which includes a total of 6,060 features [14].

For Viola and Jones' approach, training time for the entire 38 layer detector was on the order of weeks on a single 466 MHz AlphaStation XP900. Utilizing novel "integral histogram image" and our small feature set (892) comparing with the size of Viola and Jones (45,396), our training process can be finished in two days on a single Pentium 4 CPU 3.00GHz. "Integral histogram image" saves one third times for both training and detection.

It is an original unoptimized face detection system combining our novel feature set. The detector scans the image at multiple scales and locations. And the test set is the CMU new face test set without containing images with line drawn faces. The detection rate achieves 90% with 86 false detections. The face detector can process a 256 by 377 pixel image in about 12 seconds (using a start scale of 1 and a step size of 1.5). ROC curve is shown in Fig. 5. Fig. 6 gives some typical detection results.

6 Conclusions

Fisher linear discriminant is used to project template-based histograms features for the task of face detection in this paper. We choose best features, pairs of template-based histogram along with Fisher projection orientation, by AdaBoost algorithm. The experimental results demonstrate that the selected features are very powerful to describe the face pattern. There are a number of directions for future work, including adaptive selection of histogram dimensions, extending the framework to multi-view face detection, and employing more sophisticated image preprocessing and normalization techniques.

References

1. Viola, P., Jones, M.: Robust real-time object detection. In: IEEE ICCV Workshop on Statistical and Computational Theories of Vision. (2001)
2. Murphy, K., Torralba, A., Freeman, W.: Using the forest to see the trees: a graphical model relating features, objects, and scenes. In: Advances in Neural Information Processing Systems 16 (NIPS). (2003)
3. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: The importance of good features. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (2004) 53–60

4. Wang, H., Li, P., Zhang, T.: Proposal of novel histogram features for face detection. In Singh, S.e.a., ed.: 3rd International Conference on Advances in Pattern Recognition. Volume 3687 of Lecture Notes in Computer Science., Bath, United Kingdom, Springer-Verlag (2005) 334–343
5. Wu, J., Rehg, J., Mullin, M.: Learning a rare event detection cascade by direct feature selection. In: Advances in Neural Information Processing Systems 16 (NIPS). (2004)
6. Li, S., Zhang, Z.: Floatboost learning and statistical face detection. IEEE Trans. Pattern Anal. Mach. Intell. **26** (2004) 1112–1223
7. Torralba, A., Murphy, K., Freeman, W.: Sharing features: Efficient boosting procedures for multiclass object detection. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. Volume 2. (2004) 762–769
8. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. In: European Conference on Computational Learning Theory. (1995) 23–37
9. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Machine Learning: Proceedings of the Thirteenth International Conference, Morgan Kauffman, San Francisco (1996) 148–156
10. Lienhart, R., Kuranov, A., Pisarevsky, V.: Empirical analysis of detection cascades of boosted classifiers for rapid object detection. MRL Technical Report, Microprocessor Research Lab, Intel Labs (2002)
11. Schapire, R., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning (1999)
12. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Technical Report, Department of Statistics, Sequoia Hall, Stanford University (1998)
13. Wang, H., Li, P., Zhang, T.: Novel likelihood estimation technique based on boosting detector. In: IEEE International Conference on Image Processing. Volume 3., Genoa, Italy (2005) 477–480
14. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision (2004)

Perception Based Lighting Balance for Face Detection

Xiaoyue Jiang[1], Pei Sun[2], Rong Xiao[2], and Rongchun Zhao[1]

[1] Department of Computer Science, Northwestern Polytechnical University,
Xi'an, China 710072
xiaoyuejiang@mail.nwpu.edu.cn
[2] Microsoft Research Asia, Beijing, China 10080

Abstract. For robust face detection, lighting is considered as one of the greatest challenges. The three-step face detection framework provides a practical method for real-time face detection. In this framework, the last step can employ computation extensive method to remove the false alarm and usually some de-lighting methods are done. It is complex to model the lighting variance precisely. The usually used simplified lighting model fails under non-uniform lighting conditions for the reason that it cannot account for the cast shadow, shading, and highlight, which are the main variances caused by non-uniform lighting. According to the adaptation capacity of the human vision system, we propose a perception based mapping method (PMM) to balance the influence of non-uniform lighting. Experimental results indicate that with PMM as the lighting-filter the false positives caused by lighting variance can be removed more accurately in the face detection tasks. PMM shows its outstanding performance especially under the extreme lighting conditions.

1 Introduction

Face detection has been well recognized as a challenging problem in the vision community. Due to variations caused by pose, expression, occlusion, lighting and illumination, the distributions of face objects are highly nonlinear, and thus make the learning extremely difficult. Among these variations, light/illumination and pose changes are regarded as the most critical factors for robust face detection. Recently, view-based framework has been widely applied to reduce the variances caused by pose changes[1],[2]. However, problems caused by lighting or illumination are less addressed according to the literature.

The challenges of de-lighting for detection mainly focus in the following aspects. Firstly, there is no prior knowledge, e.g. the 3D information, except the image to be detected itself. Therefore the lighting models that are successfully applied in face recognition tasks[3],[4] cannot be applied in face detection tasks. Secondly, the de-lighting work, as the preprocessing step to every candidate test window, will bring a huge amount of computation. It is not affordable in real-time detection systems. Rowley[1] proposed a linear de-lighting algorithm as well as a histogram equalization algorithm to alleviate lighting and contrast variations

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 531–540, 2006.

in detection systems. However, several problems need to be addressed. Firstly, the lighting map estimated for a plane is just a rough approximation of the face lighting map. Secondly, subtracting the lighting map from the face deviates from the reflection function[5]. Thirdly, as the pre-filter of the detection procedure, the de-lighting algorithm introduces much computation cost for real-time detection system. To enable the rapid face detection, Xiao[6] proposed a three-step face detection framework. By dividing the detection procedure into three steps (pre-filter, boosting filter and post-filter), the computation extensive algorithm can be applied in the post-filter step. In this approach, linear de-lighting and histogram equalization algorithms are used in the post-filter step to reduce the lighting variations. However, this de-lighting approach still suffers from the performance inefficiency similar to Rowley's approach except the computation cost to the system. With the assumption that lighting is slowly changed information the quotient image (QI) based methods [7]∼[9] take the low-pass version of original image as the lighting map. The original image divided by the lighting map is the so-called intrinsic image. Roughly separating the lighting and reflectance with a fixed threshold of the frequency is not very suitable. That will make the intrinsic image lose the low frequency information of the reflectance, which contributes to the detection task, and some high frequency information enlarged greatly, which contains the original noise in the image and the abrupt changes of the lighting. This is the halo effect of QI.

Due to the complexity of modeling the lighting variance we proposed a method to balance the lighting influence motivated by the adaptation of human vision system (HVS). For an image taken under non-uniform lighting conditions, we should make it rich of details and in a proper brightness, intuitively. That is the image should be adjusted according to the local as well as the global situation. To decide the adjustment parameters we apply the intensity entropy as the assessment of the adjusted images.

The rest of the paper is organized as following. In Sec. 2, we present the perception based lighting balance method and its application in the face detection system. Sec. 3 compares some different lighting adjustment methods in face detection on different face databases. The conclusions are drawn in Sec. 4.

2 Perception Based Mapping

In this section we will first introduce the relationship of the lighting and intensity value in the images. Then we apply the adaptation model of HVS to adjust the lighting conditions of the images. At last the parameters are optimized according to the assessment of the images.

2.1 Background

According to the reflection function

$$I(x,y) = \int_{\Omega} \rho(x,y) L_i(x,y) \cos\theta_i d\omega \tag{1}$$

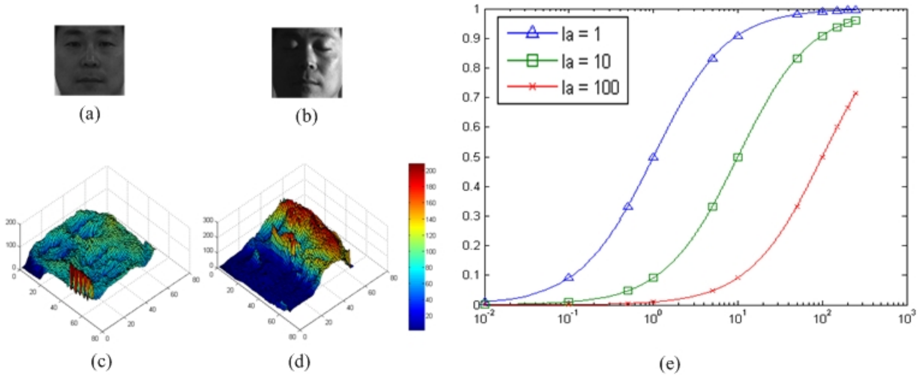


Fig. 1. (c)and(d) are the mesh figures of (a)and(b),respectively.The mesh figures indicate that the lighting influences the dynamic range of the images. (e)The curves of perception mapping function (the x-axis is the value of I and y-axis is the value of V).

The intensity $I(x, y)$ of the point (x, y) is decided by the reflectance $\rho(x, y)$ and all the effective incident lighting $L_i(x, y) \cos\theta_i$ on that point. We can rewrite it in the discrete form,

$$I(x, y) = \sum_i R(x, y) \times L_i(x, y) \cos\theta_i = R(x, y) \times L(x, y) \qquad (2)$$

where $L(x, y) = \sum_i L_i(x, y) \cos\theta_i$ is the sum of all the effective lighting for that point, and $R(x, y)$ is the reflectance character of the point. According to the discrete reflection function (equation 2), lighting takes charge of perception gain. As seen in Fig.1(c) and (d), the non-uniform lighting suppresses the features in the shadow region and exaggerates the features in the highlight region.

2.2 Mapping Function

The accurate adaptation of HVS is achieved through the cooperation of mechanical, photochemical, and neural processing in the vision system. According to the results from electro-physiology, the photoreceptor is the crucial element that takes charge of the procedure of adaptation. The photoreceptor can be modeled as the function of intensity as follows,[10][11]

$$V = \frac{I}{I + \sigma(I_a)} V_{max} \qquad (3)$$

where V is the potential produced by cones; I is the input intensity; and V_{max} determines the maximum range of the output value. Function $\sigma(I_a) = (f I_a)^m$ takes the role of the semi-saturation constant, i.e. when $I = \sigma(I_a)$, the output value is half of the input intensity. Since the new image should be in the range of [0,255], we set the maximum range $V_{max} = 255$.

In Fig.1(e), the curves are drawn using the log linear scale. The curves are the value of perception mapping function with $m = 1$, $f = 1$ and $V_{max} = 1$. Although

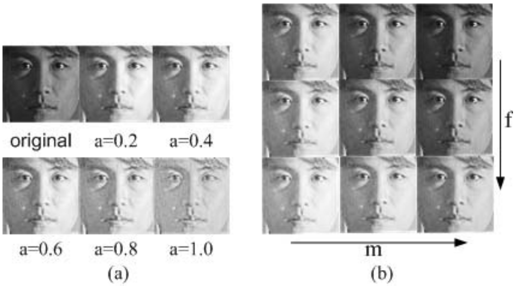


Fig. 2. (a) coefficient α controls the detail information (b) m and f control the contrast and brightness respectively

I_a is variant, the shape of the curves still keeps more or less the "S" shape. This behavior is close to old photographic transfer functions, and is also a defining characteristic of parts of human vision. These functions are fitted for electrophysiological measurements of photoreceptors and concluded from psychophysical experiments. At the same time the curves satisfy the requirement of adjustment for the images. For the darker region, the absolute intensity of pixels and the contrast will be enhanced, while the difference between the pixels in the highlight part will be suppressed. As a result, the image is adjusted to a better situation.

Adaptation Level I_a. If we choose the average intensity of the image as the adaptation level I_a, the adjustment is global. It does not perform any specific processing to the darker or brighter region and some details in those regions may be lost. To compensate the details, the local conditions of every point should be considered. We can use the bi-linear interpolation to combine the global adaptation I_a^{global} and local adaptation $I_a^{local}(x,y)$ as,

$$I_a(x,y) = \alpha I_a^{local}(x,y) + (1-\alpha) I_a^{global} \tag{4}$$

$$I_a^{local}(x,y) = K(I(x,y)) \tag{5}$$

$$I_a^{global} = mean(I) \tag{6}$$

Different kernel $K(\bullet)$ can be applied to extract the local information. Gauss kernel is the most commonly used one. The interpolation of the global and local information will adjust the details. In Fig.2(a), with the increasing of the parameter α, the details become notable gradually. When $\alpha = 1$, i.e.$I_a = I_a^{local}$, all the details are expressed out including the noise.

Parameter f and m. The other two parameters f and m control the intensity and contrast, respectively. Parameter f is the multiplier in the adaptation function, i.e. to every point's adaptation level $I_a(x,y)$, f magnifies them on a same scale. The brightness of the whole image will be enhanced or suppressed accordingly. The alternation of brightness can be shown only when changes on f is large enough. In [11], the parameter f is suggested to be rewritten in the following form

$$f = exp(-f') \tag{7}$$

With a comparative smaller changing range of f', f can alter the brightness of the image. The parameter m is the exponent in the adaptation function. Different from the parameter f, m magnifies every $I_a(x, y)$ on a different scale based on adaptation value. Therefore, the parameter m can emphasize the difference between every point, i.e. the contrast. In Fig.2(b), the parameter α is fixed. With the increment of m, the contrast of the image is enhanced in every row. And in every column, the brightness of the image is enhanced with the increase of f.

2.3 Image Assessment

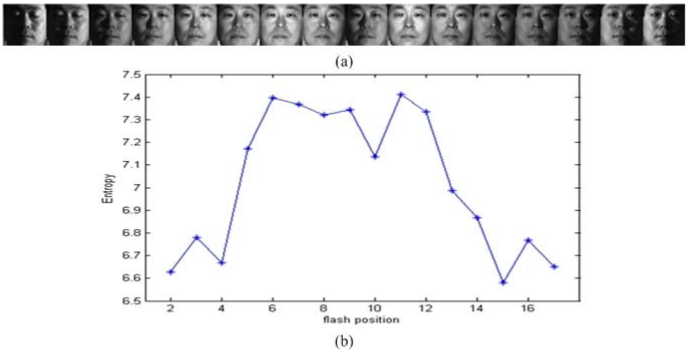
We need a comparatively objective standard to evaluate the lighting conditions of the image so that we can optimize the parameters of mapping function. An image with larger entropy means the distribution of intensity is more unified, i.e. every different intensity value has almost the same probability to appear in the image. Consequently the image will be rich of smooth changes. In vision the image will be abundant of details and without abrupt noises. Therefore, we evaluate the image with its intensity entropy.

$$H(X) = -\sum_{x=0}^{255} p(x) log_2(p(x)) \tag{8}$$

where $p(x)$ is the probability of the intensity value that appears in the image.

We take 65 sets of images in PIE database[12]. Each set of the images are taken with flashes in 16 different positions. The sample of images is shown in Fig.3(a). The mean value of the entropy with different flash position is given in Fig.3(b). We can see that the more uniform the lighting conditions are, the larger the entropy value is. The images taken with flash 6 and 11 have the maximum entropy, and in vision, the lighting is most uniform in those images.

We should point out that the histogram equalization (HE) will make the histogram flatter in theory. However, it may lose information instead of adding



(a)

(b)

Fig. 3. (a) Example of face images under different lighting conditions (from left to right the flash number is from 2 to 17); (b)The mean value of the entropy for images taken under different lighting

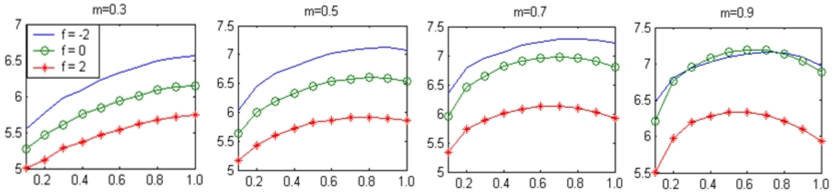


Fig. 4. The entropy curves with different f' and m. (x-axis is the parameter α and y-axis is the entropy.)

new information to the image. Thus, the resultant image will always show abrupt changes and lose details. Therefore, an image through histogram equalization will not increase the entropy but always reduce it.

2.4 Parameters Optimization

Applying the entropy as the standard to evaluate the image, we can choose a set of optimized parameters. In Fig.4, we show a set of entropy curves of the same image corrected with different parameters. As illustrated in the figure, with the increasing of m, the entropy value is enlarged regardless of f' and α. From every subfigure, we can find that the parameter α controls the shape of the curves regardless of f'. Based on the above observations, we can first fix m and f' to optimize α. That is we first find the point can combine the local and global adjustment best. Accordingly the information containing in the image can be presented to a full extent. Then with α_0 (the optimized α) we adjust the dynamic range and contrast of the image, i.e. to optimize the parameter f and m, respectively.

The initial estimation of m is given based on the key of the image. The image key value k can be estimated using the log average L_{av}, log minimum L_{min}, and log maximum L_{max} luminance,

$$k = \frac{L_{max} - L_{av}}{L_{max} - L_{min}} \tag{9}$$

Then m is chosen as,

$$m = 0.3 + 0.7^{m} \tag{10}$$

This function is based on extensive experiments and it makes the value of m in the range reported by electro-physiological researches[11]. For the parameter f', we usually set $f' = -2$. With the estimated m and f, we can optimize the parameter α first. α_0, the optimized α, will make the image entropy maximum.

With α_0 and estimated m, we optimize the parameter f' as

$$min_{f'} |H(I(\alpha_0, m, f')) - T| \tag{11}$$

where T is the expected entropy value. We do not choose the parameter f' with the maximum entropy. The reason is that with the increasing f', the brightness is increased. So we set a threshold to hold the brightness, e.g.$T \in [7.0, 7.5]$. Finally, with α_0 and f'_0 (the optimized f') we adjust the parameter m to make the image entropy maximum.

3 Experiments

In the three-step face detection framework [6], the classifier in the first step is designed to be simple. It can reject negative samples with little computation over two to three features. The classifier of the second step is designed to be efficient. It should reduce the false positive (FP) rate to the scale of 10^{-7} with as little computation as possible. And the classifier of the third step is designed to be accurate. It should remove the FPs precisely. To test the effect of the de-lighting methods on the face detection, we only train the face detector with the third step classifier. And the de-lighting method is designed as pre-filter of the classifier. First, we do de-lighting for the images. Then, we extract the Gabor feature of the adjusted images and choose the most discriminative features that can decide whether the image is face or non-face through boosting method. With different pre-filters, we train four face detectors. The four pre-filters are PMM, HE, QI and none de-lighting, respectively.

3.1 De-lighting Results

In Fig.5, we compare the results of these de-lighting methods and their edge images. Edge is the representation for the local contrast. It can be applied to express the effect of the de-lighting methods. As illustrated in Fig.5, the PMM has some great advantages. In vision, it can module the images to a better condition. The effect of the non-uniform lighting is weakened and the appearance of the face is kept. From the edge images we can see that the PMM recoveries the details of the face, especially the part in the deep shadow, e.g. the left eye. Although HE can also do the recovery, it brings much abrupt noise. QI removes the low-frequency information and makes the appearance of the face destroyed. It also magnifies the original noise in the image, such as the abrupt changes of the lighting on the nose.

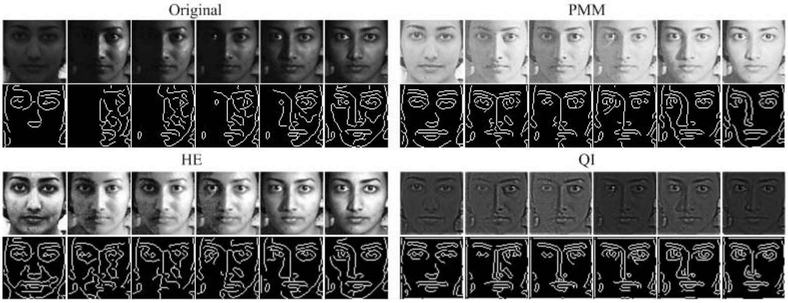


Fig. 5. The de-lighting results of different methods and their corresponding edge images

3.2 Comparison of Detectors

Training Detectors. More than 12000 images without faces and 10000 face images were collected by cropping from various sources, such as AR, Rockfeller, FERET, BioID and the WEB [13][14]. Most faces in the training set have variations in terms of pose and lighting. A total number of about 80000 face samples with the size of 32 × 32are generated from the 10000 face images by random transformation: mirroring, four-direction shift with 1 pixels, in-plane rotation within 15 degrees, and scaling within 20% variations. 20000 face samples are chosen randomly for training the face detectors.

Table 1. Illumination condition of every PIE subset

	subset1	subset2	sebsut3	subset4	
Ambient Lighting	Yes	Yes	No	Yes	No
Flash No.	None	4∼22	5∼18	2,3	2∼4 and 19∼22

Table 2. Detection results on the PIE subsets

	subset1	subset2	subset3	subset4
Raw	436/2	1266/26	893/22	535/63
HE	438/0	1292/0	914/1	591/7
QI	435/3	1289/3	913/2	585/13
PMM	438/0	1292/0	914/1	598/0

Testing Detectors. 1) PIE Test Set: We separate the frontal face of PIE (c27 serial) into four sets based on the lighting conditions. The details of every set are shown in Table 1. The detection results of the detectors with different pre-filters on these data sets are shown in Table 2. (Raw means do not do delighting; the value X/Y in the table means True Positive/False Negative) As seen in Table 2, the performances of these detectors are almost same when the objects are under nearly uniform lighting conditions. Under those conditions, the commonly used assumption that lighting is the low-frequency information and the object is the convex Lambertian is tenable. Based on this assumption, these methods can alleviate the lighting influence. Under the extreme lighting conditions (e.g. set 4), the lighting's real influence on the non-strict convex object is shown and the traditional methods do not work under that situation. Under such an extreme lighting condition, PMM can still keep its performance. The PMM is derived from the mechanism of photoreceptor that is adaptable to a wide range of lighting conditions. The global and local information are both used to adjust the image so that the image can be presented in a better condition. As a result, the non-uniform lighting influence is balanced. As the pre-filter in the face detector, PMM achieves its task to accurately remove the FPs caused by the lighting variance.

2) Composite Test Set: As we mentioned before, about 80000 face samples are generated from 10000 face images, out of which 9024 faces and 9315 non-face

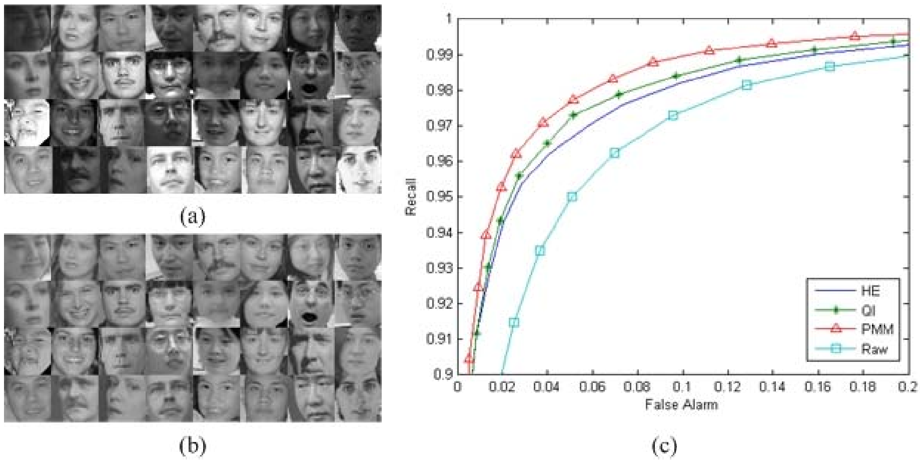


Fig. 6. (a)some sample of composite test set. (b)the PMM de-lighting results of (a). (c)ROC curves of different detectors.

images are chosen randomly to test the detectors. Fig.6(a) gives some examples of the test images and(b)are the PMM results of (a). Fig.6(c) shows the ROC curves of the detectors. Comparing the ROC curves of different detectors, we can see that the performance of the detector using PMM as the pre-filter is better than the others. There are not so many images taken under extreme lighting in the test set. Therefore, the improvement made by PMM filter is not so great. However, for the lighting correction filter in the detection framework, it needs to eliminate the effect caused by lighting whatever the lighting is. PMM shows that it can discard the FPs caused by lighting variance more precisely than others. This characteristic of PMM is necessary for the lighting-filter in the three-step face detection framework.

4 Conclusion

To model the lighting variance is very complex . The simplified lighting model only works under uniform lighting conditions and will fail under non-uniform lighting conditions. Therefore, we do not try to model lighting with the rough model. Motivated with the super adaptation of HVS to different lighting environment, we introduce the perception based mapping methods to eliminate lighting variance. Taking the entropy as the image assessment criterion, we optimize the parameter for the mapping function. PMM can balance the non-uniform lighting influence and modulate the image to a better situation. Even under extreme lighting conditions, PMM still can keep its performance. As the lighting-filter, PMM removes FPs caused by lighting variance more precisely than other methods in the face detection experiments.

PMM does not require the prior knowledge of the subject. Thus the PMM also can be applied to adjust the lighting conditions of images about other subjects except face. It can serve as the lighting-filter for other image processing algorithms.

References

1. Rowley, H., Baluja, S., and Kanade, T.: Neural network based face detection. IEEE Trans. Pattern Analysis and Machine Intelligence 20(1), (1998)22-38
2. Schneiderman, H. and Kanade, T.:A Statistical Method for 3D Object Detection Applied to Faces and Cars. IEEE International Conference on Computer Vision,(2000)746-751
3. Basri, R. and Jacobs, D.: Lambertian reflectance and linear subspaces, IEEE Trans. Pattern Recognition and Machine Intelligence, 25(2),(2003)218-233
4. Georghiades, A., Kriegman, D. and Belhumeur, P.: Illumination cones for recognition under variable lighting: faces, IEEE Conf. Computer Vision and Pattern Recognition, (1998)52-58
5. Ramamoorthi, R. and Hanrahan P.: A Signal-Processing Framework for Inverse Rendering, Proceedings of the 28th annual conference on Computer graphics and interactive techniques, (2001)117-128
6. Xiao, R., Li, M., and Zhang, H.:Robust multipose face detection in Images. IEEE Trans. Circuits and Systems for video technology, 12 (1),(2004)31-41
7. Jobsom, D., Rahaman, Z. and Woodell, G.:A multiscale retinex for bridging the gap between color images and the human observation of scenes. IEEE Trans. Image Processing, 6(7), (1997)965-976
8. Gross, R. and Brajovic,V.: An image processing algorithm for illumination invariant face recognition. 4th International Conference on Audio and Video Based Biometric Person Authentication, (2003)10-18
9. Wang, H., Li, S., and Wang, Y.: Generalized quotient image. IEEE Conf. Computer Vision and Pattern Recogniton (2004)498-505
10. Reinhard, E., Stark, M., Shirly, P. and Ferwerda, J.:Photographic tone reproduction for digital images. ACM Trans. Graphics, 21(3), (2002)267-276
11. Reinhard, E. and Devlin, K.:Dynamic range reduction inspired by photoreceptor physiology. IEEE Trans. Visualization and Computer Graphics, 11(1), (2005)13-24
12. Sim, T., Baker, S., and Bsat, M.: The CMU Pose, Illumination, and expression (PIE) database. Processing of the IEEE International Conference on Automatic Face and Gesture Recognition, (2002)
13. Martinez, A.M. and Benavente, R: The AR Face Database. CVC Technical Report #24, (1998)
14. Phillips, P.J., Wechsler, H., Huang, J., and Rauss, P.J.: The FERET database and evaluation procedure for face-recognition algorithms. Image and Vision Computing, 16(5), (1998)295-306

An Adaptive Weight Assignment Scheme in Linear Subspace Approaches for Face Recognition

Satyanadh Gundimada and Vijayan Asari

Department of Electrical and Computer Engineering, Old Dominion University,
Norfolk, VA23529, USA
{sgund002, vasari}@odu.edu

Abstract. A methodology for determining the level of confidence of a subregion in the overall classification of a given face image affected due to varying expressions, illuminations and partial occlusions is presented in this paper. The technique for obtaining the weights for each individual region of the test image is based on a measure of optical flow between that test image and a face model. Individual image regions or the modules are also assigned additional weights by arranging them in the order of their importance in classification. The approach presented is applicable mainly in scenarios where the number of samples in the training set is too little. A K-nearest neighbor distance measure is used in classifying each module of the test image after dimensionality reduction. A total score is calculated for each training class based on the classification result of each module and its associated weights. Considerable increase in recognition accuracy has been observed for PCA, LDA and ICA based linear subspace approaches when implemented using the proposed technique.

1 Introduction

Over the past 15 years, research has focused on making face recognition systems more accurate and fully automatic. Significant advances have been made in the design of classifiers for successful face recognition. Among appearance-based holistic approaches, eigenfaces [1, 2] and Fisherfaces [3] have proved to be effective on large databases. PCA performs dimensionality reduction by projecting the original n-dimensional data onto the lower dimensional linear subspace spanned by the leading eigenvectors of its covariance matrix. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pair-wise de-correlated. Unlike PCA, LDA encodes discriminating information in a linearly separable space using bases that are not necessarily orthogonal. Kernel methods such as Kernel Principal Component Analysis (KPCA) and Kernel Fisher Discriminant Analysis (KFDA) [4] show better results in face recognition than linear subspace methods. Nonlinear projection based methods have been able to overcome the problem of expressions and lighting in face images to some extent. But there has not been a significant improvement in the recognition accuracy in situations where the face images undergo lot of variations including expressions, partial occlusions and lighting variations and at the same time not many samples are available to represent the distribution. This paper implements a method of face recognition which is based on locally weighted regions. The weights assigned to

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 541 – 550, 2006.

the image regions enhance the recognition accuracy by decreasing confidence in the modules which are affected due to facial variations. There are recent publications [5-7] in this direction of expression, occlusion and lighting invariant face recognition. In [6] a weighted distance measure is implemented which reduces the effect of pixels in the test image which underwent significant movement from the corresponding positions in the training images caused due to expression variations.

The technique presented in this paper is aimed to deal with varying expressions, partial occlusions and extreme lighting variations. It also implements a technique in which the regions which are highly affected due to facial variations are automatically located and the results of classification of such regions are replaced with the classification result of the corresponding regions on the other half of the test image provided those regions have been determined to be less affected due to facial variations. This is implemented by assuming that the test image is a frontal face image, with proper alignment with the training images. A technique based on the optical flow between the test image and a face model is implemented to determine the weights associated with the modules of the test image. The weight assigned to each module is proportional to the sum of magnitudes of the optical flow vectors within that module. An overall score is calculated taking into consideration the classification result of each of the modules and the weights associated with that module after replacement procedure to determine the final result of classification of the test image. The technique presented here is computationally efficient and has achieved very high accuracy rates on certain freely available standard face databases. The testing strategy is implemented such that the training set consists of face images taken under controlled conditions where as the testing set consists of images captured in uncontrolled conditions. The paper is organized as follows. The second section describes the effect of facial variations on recognition accuracy and the role of modularization in reducing those effects. Third section provides the implementation steps of the proposed technique. Section four explains the details about the type of testing strategy that is implemented along with the obtained recognition accuracies on various databases.

2 Variations in Face Images

Variations caused in facial images due to expression, makeup and non uniform lighting tend to move the face vector away from the neutral face of the same person both in image space and reduced linear subspace. It has been observed that the dimensionality reduction techniques on individual modules of the face images improve the accuracy of face recognition compared to applying on the whole image. An experiment is conducted on AR database to show the effect of modularization of the face images on recognition accuracies. Two sets of images, one with expressions mostly affecting the mouth regions and the other set with partial occlusions on the bottom half of the face images. The two sets are tested separately using a leave one out strategy. All the training images are divided into 64 local regions. Each region or the module is projected into a reduced eigen space. The test module is classified using a nearest neighbor algorithm. The accuracies of the individual modules for both the sets are shown in figure 1.

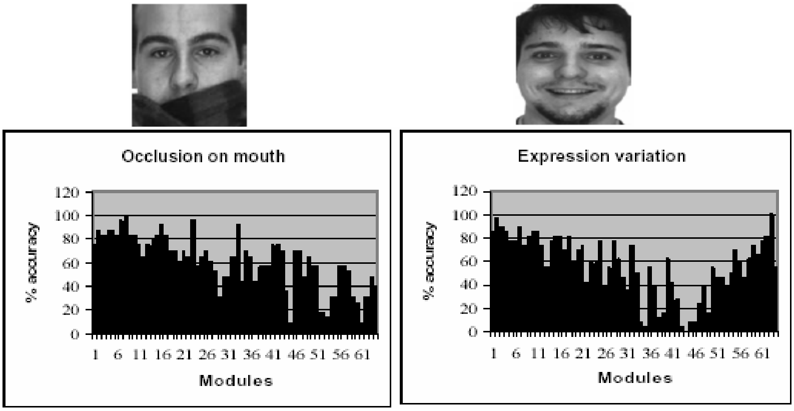


Fig. 1. Percentage of accuracies of each module in the images affected due to partial occlusion, expression variation

Sample face images of each set are also displayed besides the accuracy results in the figure 1. It can be observed that only the regions that are affected adversely due to facial variations have low accuracy rates. Hence it can be said that in most cases it is still possible to obtain good accuracy results if we are able to find out the amount of variation on a local region of the face image.

3 Weighted Modules

In many of the face recognition techniques [8] which are based on segmented human face regions, each module or the local region of the test image is classified separately and the overall classification of the test image is determined by employing a voting mechanism on the classification results of all the individual modules. The classification is done in favor of the class or individual who obtains the maximum votes. Instead of a voting technique to classify the test image, a weighted module approach is implemented in this paper. Initially images in the face database are divided into a predefined set of modules, ‘m’. These modules are arranged in the order of importance of classification using the first set of weights as given in [10]. The directions of maximum variance are then calculated for each module separately using principal component analysis and the weights are obtained after projecting the vectorised modules of each training image onto their respective subspaces. The algorithm presented in this paper differs from the earlier works in using the image modules more effectively for the overall classification of the image. A second set of weights are assigned to the modules dynamically. A less weight or confidence is given to those modules of the test image which are affected due to the variations caused because of expressions, makeup or decorations, occlusions, and lighting. Determination of the weights associated with each module is achieved by the application of an optical flow algorithm between the test image and a face template.

3.1 Optical Flow

Optical flow between the test image and a neutral face template is calculated to determine the regions with expressions, partial occlusions, and extreme lighting changes. The face model which is used as a reference image for a neutral face is the mean of all the face images in the training database. Lucas and Kanade's algorithm [9] is a classical technique and is implemented to find the optical flow between the test images and the face model in this paper. A brightness constancy constraint is assumed in the calculation of optical flow as given in equation 1.

$$I(x, y, t) = I(x + u\delta t, y + v\delta t, t + \delta t) \tag{1}$$

I(x,y,t) is the intensity at the image pixel *(x,y)* at time *t*. *(u,v)* is the horizontal and vertical velocities, δt is a very small time interval. Tailor series expansion of equation 1 results in the optical flow constraint as given in equation 2.

$$u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \tag{2}$$

An additional smoothness constraint [9] is assumed to solve for the velocity vectors at each pixel of the image. The test images are preprocessed before applying the optical flow algorithm. Each of the test images is passed through a low pass filter to reduce the resolution of the image. This step helps in ignoring the smaller variations between the test image and the face model. The optical flow algorithm which is applied after low pass filtering the image captures the variations which are prominent in both the images, such as expressions, decorations, and occlusions.

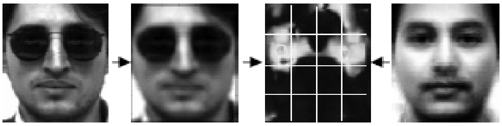


Fig. 2. Left to right are the test image, low pass filtered image, optical flow magnitude image and the face template

We are only interested in the magnitudes of the optical flow vectors and not the directions of the vectors at each pixel. It can be observed in the figure 2 that the regions which have undergone a lot of variation with respect to the face model have a higher sum of magnitudes of optical flow vectors. It is assumed that the face images are properly aligned prior to finding optical flow. Any changes in alignment could produce false variations.

3.2 Assignment of Weights

There are two sets of weights that each individual face module is associated with. The first weight assignment scheme is based on the importance of each module for the overall face recognition [10]. The assumption is that the various facial regions have different amounts of importance, whereby the eye and mouth regions play important role in face recognition [10]. A weighted function defined according to the spatial position of the respective regions of the facial features is used in this paper. The second weight assignment scheme is based on the optical flow magnitudes within each module. The modules enclosing the regions with higher variations are assigned lesser weight in order to minimize the influence of such modules on the overall accuracy of classification. Equations 3 and 4 explain the process of assignment of weights.

$$w_i = \frac{(G_{\max} - G_i)}{G_{\max}} \quad for\ G_i > T \tag{3}$$

$$w_i = 1 \qquad for\, G_i \le T \tag{4}$$

where G_i is the sum of the magnitudes of the optical flow vectors within each module. T is the magnitude threshold, below which the module is given full confidence during classification. T is specific to each module and is set in such a way that variations that exist within the neutral face images i.e., without any expressions or decorations or lighting variations are not penalized.

$$G_k = \sum_{p=1}^{(N/m)} \sum_{q=1}^{(N/m)} \left\| F_{pq} \right\| \qquad for\, k = 1,2,3,.....m \tag{5}$$

where $\|\mathrm{Fpq}\|$ is the magnitude of the optical flow between the test image and the average face template at pixel *(p,q)*.

$$G_{\max} = Max(G_k)_{\forall k} \tag{6}$$

The weights are set in such a way that the modules that enclose maximum variations are given zero weightage or no confidence. The modules whose flow magnitude does not exceed threshold T are given a weightage of ‘1’ and the rest of the modules that lie between the limits are assigned weights according to equation 3. Figure 3 shows a graphical representation of the assignment of weights.

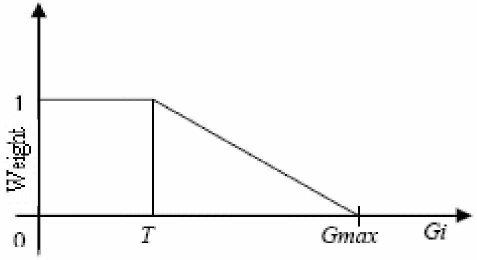


Fig. 3. Illustration of the linear weight assignment policy

3.3 Threshold Calculation

Threshold ‘*T*‘ represents the magnitude of variations below which the weights assigned is always ‘1’. The variations below this threshold are not considered as the ones that are caused by expressions, occlusions and non uniform lighting. The mean image of the set of face images belonging to the same individual is selected to represent the neutral face of that individual. Then optical flow is calculated on the mean image of each individual and the face template. Sum of the magnitudes of flow vectors within each module is calculated. Maximum magnitude obtained for each module over all the mean images is taken as the threshold for that module. Equations 7 and 8 further explain the procedure of calculating threshold '*T*'.

$$G_{nr} = \sum_{i=1}^{(N/m)} \sum_{j=1}^{(N/m)} \left\| F_{ij} \right\| \qquad for\ r = 1,2,3,.....m \quad and \quad n = 1,2,....N \tag{7}$$

$$T_r = Max(G_{nr}) \qquad where\ \ n = 1,2,3,..N\ \ and\ \ r = 1,2,3,...m \tag{8}$$

G_{nr} is the summation of the magnitude of the optical flow vectors within each module for all the mean faces of the individuals with respect to the face template. T_r is the threshold for each of the module obtained by taking the maximum value of G_{nr}.

3.4 Symmetrically Opposite Modules

When very less training images are available, it is not possible to easily determine the intra personal subspace probability distribution for each module. In such cases it is observed that replacing the result of classification of a module which had undergone higher variations in comparison with the corresponding module on the other half of the symmetric frontal face image which had undergone lesser variations would produce better results.

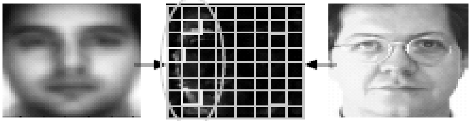


Fig. 4. Classification result of the regions encircled are replaced with the result of the symmetrically opposite modules

The classification result of a module is replaced by the other only when the difference between the magnitudes of variations exceeds beyond a certain level. This threshold is experimentally determined in order to maximize the recognition accuracy. The product of weights corresponding to the modules belonging to the same class are summed up. Which ever class receives the highest score determines the classification result of the test image.

4 Experimental Results

PCA, which is a linear subspace approach, is implemented to prove the efficiency and do the analysis of the proposed method in improving the recognition accuracy. Testing of the proposed technique is carried out on three databases, AR, AT&T and Yale individually. 40 individuals are chosen randomly from the AR database. 13 images of each individual are present in this database. The AT&T database has 40 individuals with 10 images of each. Yale database has 150 images in total with 15 individuals. Two types of testing strategies are implemented here. The first one is the classical leave one out technique. In the second testing policy, only 4 images of each individual are chosen to form a training set. For example in AR database, out of the 13 images corresponding to each individual, 4 images are selected to form the training set. The rest of the 9 images form the testing set. Similarly when dealing with Yale database, out of the 11 images of each individual, 4 are assigned for training and 7 for testing. This represents the real life situation where the training set consists of images taken in controlled environments where as the probe images are uncontrolled. The images selected for training are closer to the neutral face of the individuals. The sample images for both the training and testing from the AR database are provided in figures 5 and 6 respectively. probe image is aligned with the face model to eliminate the possibility of false motion between the two images due to misalignment. A second step is to establish a dense correspondence between the two face images using optical flow technique and then to calculate the magnitudes of the flow vectors. The probe image is

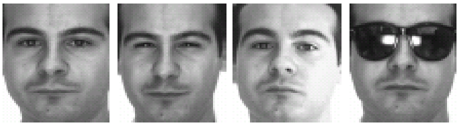


Fig. 5. Face images of an individual in the training set from the AR face database

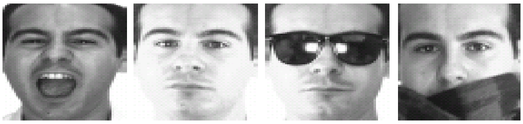


Fig. 6. Some of the face images of a individual in the testing set from the AR face database

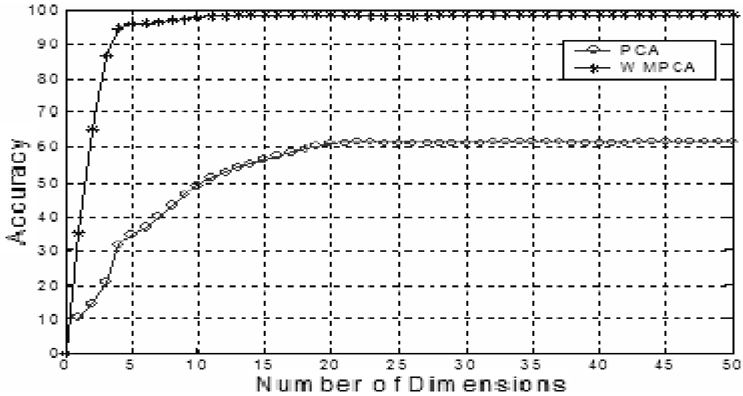


Fig. 7. Accuracy vs the number of dimensions of the subspace corresponding to AR database for testing strategy 1

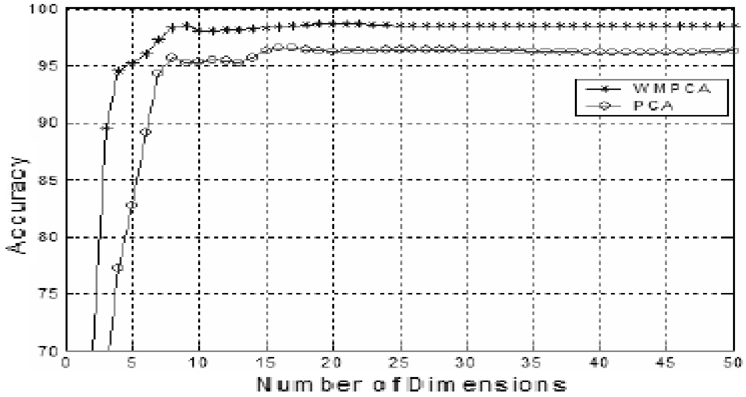


Fig. 8. Accuracy vs. the number of dimensions of the subspace corresponding to AT&T database for testing strategy 1

modularized and the summations of the magnitudes of the vectors within each module are calculated and the weight factor is assigned to each module. Each module of the probe image is projected onto the corresponding linear subspace created from the training set and then a K- nearest neighbor distance measure is used to classify that module. A final score for each class or individual in the training set is calculated by taking consideration of the weightage associated with each module. A winner takes all strategy is followed in determining the final classification result. Figures 7 and 8 illustrate the accuracies of the proposed technique on AR and AT&T databases respectively. Leave one out testing strategy was used for testing to obtain the results. It can be observed that even with comparatively large set of training images.

The holistic linear subspace approach is unable to provide high accuracy on AR database. On the other hand it can be observed that the accuracy results of both PCA and the proposed method (WMPCA) are closed to each other in the case of AT&T database.

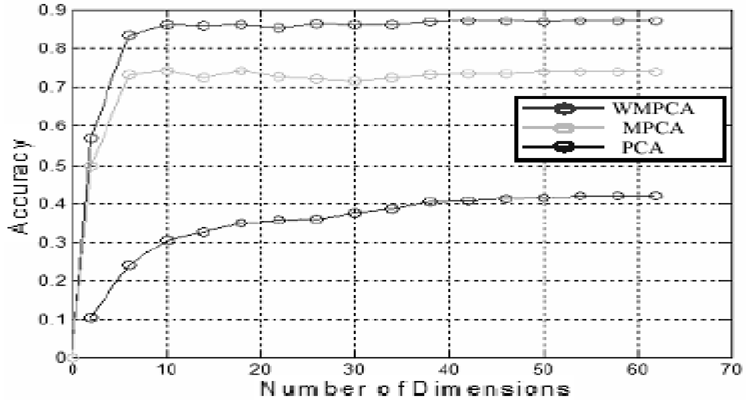


Fig. 9. Accuracy vs the number of dimensions of the subspace corresponding to AR database for testing strategy 2

This is mainly due to the fact that AT&T database has almost all the images under controlled conditions. Figure 9 and 10 demonstrate the results of three different techniques, PCA, MPCA (modularized principal component analysis with voting), and WMPCA using the second testing strategy explained above on AR and Yale databases respectively. It can be observed that the holistic PCA has failed miserably on AR database. The modularized PCA approach with voting did provide better results but still around 12 % less accurate when compared to that of the proposed technique. The same explanation can be attributed to the obtained results i.e., uncontrolled test images when compared to controlled training images. The difference in accuracy levels between the three methods in the case of Yale database shown in figure 10 is not so prominent due to the fact that the variations are less between the training and the test images.

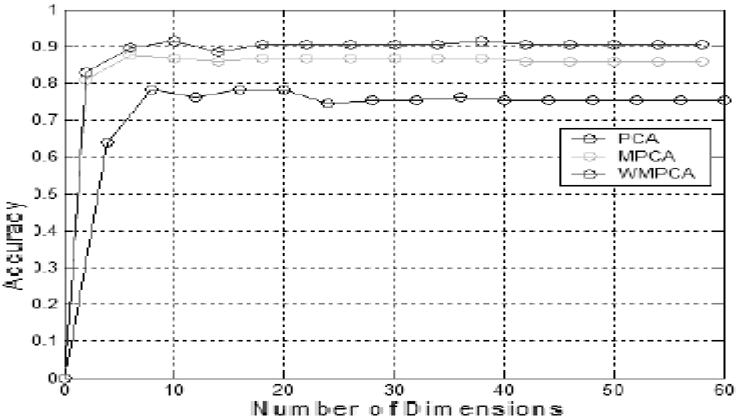


Fig. 10. Accuracy vs the number of dimensions of the subspace corresponding to Yale database for testing strategy 2

Table 1. Accuracy of PCA, LDA , ICA methods vs proposed technique on AR and AT & T databases

Database	% Accuracy					
	PCA	ICA	LDA	Proposed method		
				PCA	ICA	LDA
AR	61.15	63.75	70.32	98.85	97.88	98.9
AT&T	96.00	96.72	97.87	99.25	99.12	99.42

Table 1 gives the recognition accuracies of holistic PCA, ICA, LDA and that of the proposed weighted local regions method with the same linear subspace approaches for 64 modules. It is evident that the accuracy levels do improve than the conventional implementation of these techniques. Leave one out testing strategy was implemented to obtain the demonstrated results in the table.

5 Conclusion

The paper presented an efficient methodology of estimating the amount of variations on the local regions of a face image due to varying expressions, non uniform lighting and partial occlusions. A weight assignment scheme which assigns a proportional weight to account for the variations is also implemented. An additional weight which is based on the importance of the module in classification has also been assigned to individual modules. Classification of the test image was carried out by taking into consideration the result of classification of each module of the test image along with the associated weights with that module. Recognition accuracies provided in the paper using different testing strategies demonstrated the capability of the proposed technique in comparison with other conventional methods.

References

1. Turk, M., Pentland, A.: Eigenfaces for Recogni-tion. Journal Cognitive Neuroscience, (1991)
2. Pentland, A., Moghaddam, B., Starner, T.: "View-based and modular eigenspaces for face recognition," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (1994) 84-91.
3. Zhao, W., Krishnaswamy, A., Chellappa, R., Swets, D, L., Weng, J.: Discriminant Analysis of Principal Components for Face Recognition. Int. Conference on Auto-matic Face and Gesture Recognition 3 (1998) 336-341.
4. Huang, Jian., Yuen, Pong., Chen, C., Sheng, Wen., Lai, J, H.: Kernel subspace LDA with optimized kernel parameters on face recognition. IEEE International Conference on Automatic Face and Gesture Recognition, (2004) 327-332.
5. Martinez, Aleix.: Recognition of partially occluded and/or imprecisely localized faces using a probabilistic approach. IEEE Computer Society Conference on Computer Vi-sion and Pattern Recognition. 1 (2000) 712-717
6. Zhang., Yongbin., Martinez, Aleix, M.: Recognition of expression variant faces using weighted subspaces. Proceedings of the 17th International Conference on Pattern Recognition. (2004) 149-152
7. Tan, K., Chen, S.: Adaptively weighted sub-pattern PCA for face recognition. Neurocomputing. 64(2005) 505-511
8. Lucas, Kanade.: An iterative image registration technique with an application to ste-reo vision. Proc. DARPA Image Understanding Workshop. (1981) 121-130
9. Moghaddam,B., Pentland,A.: Probabilistic visual learning of object representation. IEEE transaction on Pattern Analysis and Machine Intelligence.(1997) 696-710.
10. Lam, K.M., Siu,W.C., Yang, S.: Human face recognition based on spatially weighted Hausdorff distance. Pattern Recognition Letters. (2003) 499-507

Template-Based Hand Pose Recognition Using Multiple Cues

Björn Stenger

Toshiba Corporate R&D Center,
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki 212-8582, Japan
bjorn@cantab.net

Abstract. This paper presents a practical method for hypothesizing hand locations and subsequently recognizing a discrete number of poses in image sequences. In a typical setting the user is gesturing in front of a single camera and interactively performing gesture input with one hand. The approach is to identify likely hand locations in the image based on discriminative features of colour and motion. A set of exemplar templates is stored in memory and a nearest neighbour classifier is then used for hypothesis verification and pose estimation. The performance of the method is demonstrated on a number of example sequences, including recognition of static hand gestures and a *navigation by pointing* application.

1 Introduction

Detecting and tracking hands is a problem in computer vision, which has been receiving significant attention. The applications in the domain of human computer interaction (HCI) are appealing: Systems have been designed for sign language recognition [1, 2, 3], navigation and control by hand motion [4, 5, 6, 7], and capturing detailed finger articulation [8, 9, 10, 11, 12]. For a given recognition task, it is important to look at the assumptions made in each system in order to determine whether or not the method will work. Two common assumptions are that foreground segmentation is reliable and that object segmentation (the hand from the rest of the body) is feasible. Relaxing these assumptions makes the problem significantly harder, particularly when using a single CCD camera.

In this paper we consider the problem of hand tracking in an HCI application. The system first hypothesizes a number of hand locations in the image, yielding estimates of location x, y and scale s of the hand. This information can then be used in subsequent steps. The image subwindow is normalized and classified as background or one of several pre-defined hand poses. The suggested method requires little computational power thus allowing for interactive gesture recognition. The following section briefly reviews some recent work in which the problem of hand tracking and pose recognition has been addressed.

1.1 Related Work on Hand Tracking and Pose Recognition

A vision-based drawing system, working in real-time, was presented by MacCormick and Isard [6]. The 2D shape of a pointing hand parallel to the image

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 551–560, 2006.

plane was tracked using a top-down camera. The system used a particle filter together with colour segmentation and background subtraction. Recognition systems that work in more general settings typically discriminate between few poses only. For example, Triesch and von der Malsburg [7] used Gabor-jets as features in a classification task to discriminate between ten hand poses in visually complex scenes. For each hand pose an elastic graph was built, in which each node contained a Gabor-jet. These graphs were then matched to each frame independently in a coarse-to-fine fashion. Bretzner *et al.* [4] took a scale-space approach to locate a hand parallel to the image plane and to distinguish between five different poses. Wu and Huang [13] recognized a number of discrete poses using a learning approach, where a labelled data set is combined with a large unlabeled data set. Kölsch and Turk [5] combined tracking with detection in a real-time interface for a wearable camera system: The global hand position was estimated by tracking a large number of skin-coloured corner features. The detection system was able to detect a particular pose at a fixed orientation using cascaded classifiers. Such classifiers were also used in the sign language recognition system presented by Kadir *et al.* [1]. However, the background in the sequences shown was relatively simple and the emphasis was placed on reliable high-level gesture recognition. Finally, a number of methods have been suggested for the task of 3D hand pose estimation from a single view: A 3D geometric model is used to generate a large number of 2D appearances, which are stored in a database [8, 10, 11]. This makes the collection of a large training data set unnecessary. In many applications, however, it is not necessary to capture the pose at each frame, but only to detect certain types of poses.

2 Subwindow Localization

This section proposes an efficient method for hypothesizing a number of locations and scale (x, y, s) of hands based on colour and motion features with no prior knowledge. This is done by searching for maxima in scale-space of the different feature likelihoods. The principle is similar to multi-scale blob detectors that look for scale-space extrema, e.g. using a Laplacian or difference of Gaussian

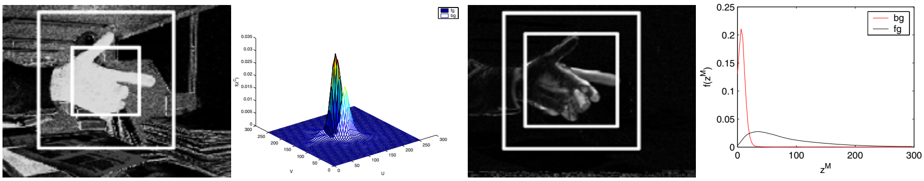


Fig. 1. Discriminative features for localization. (left) Colour and **(right)** motion features are used to hypothesize hand regions. The feature distributions of foreground and background regions are used to compute likelihoods for subregions in an image, which can be done efficiently using cumulative likelihood maps.

Table 1. Run-time comparison. This table shows the run-times to generate location hypotheses by finding maxima in scale space of a likelihood map, **(top)** using DoG filters by computing a Gaussian pyramid, and **(bottom)** using box filters on the original image (measured on a 3 GHz Pentium IV). The resulting top hypotheses were the same for both methods on the test sequences. The overhead for computing the Gaussian pyramid is high, whereas the cost for the box filter grows linearly with the number of scales. In the recognition experiments 10 scales are used.

Number of scales	3	6	9	12	15
DoG filter [14]	80 ms	99 ms	106 ms	108 ms	111 ms
Box filter	12 ms	24 ms	36 ms	48 ms	61 ms

filter [4, 14]. However, instead of working on an image pyramid we use an efficient box filter. At each image location square subwindows of different sizes are placed, and each subwindow is divided into centre foreground and surrounding background region [15], see figure 1. The relative size of the regions is fixed and is 1/8 for the case of colour features and 1/4 for the case of motion features (the inside region is larger since observable motion is often around the object silhouette). The likelihood for each subwindow in eqn (1) can be computed efficiently using integral images of the likelihood maps [15, 16]. Table 1 shows a run-time comparison of generating hypotheses by computing scale-space extrema using box filters and difference of Gaussian filters [14].

A likelihood model is defined, which explains the observation in each subwindow based on a foreground and a background feature distribution. If a hand is located within the centre region of such a subwindow, surrounded by background, the correct feature distribution will be chosen for most pixels, leading to a high subwindow likelihood. The state variable $\mathbf{x}$ is given by the parameters that define the location and scale of a chosen subwindow: $\mathbf{x} = [x, y, s]$. The observation vector $\mathbf{z}$ is described in terms of image features. At each time instant, a foreground feature distribution p^{fg} and a background distribution p^{bg} are given. Let $\mathbf{z}(\mathbf{u})$ be the observation at image location $\mathbf{u} \in \mathbb{R}^2$, and $fg(\mathbf{x})$ and $bg(\mathbf{x})$ the foreground and background image regions, respectively, given by the subwindow with parameters $\mathbf{x}$. We write the log-likelihood for a subwindow as

$$\log p(\mathbf{z}|\mathbf{x}) = \sum_{\mathbf{u} \in fg(\mathbf{x})} \log p^{fg}(\mathbf{z}(\mathbf{u})|\mathbf{x}) + \sum_{\mathbf{u} \in bg(\mathbf{x})} \log p^{bg}(\mathbf{z}(\mathbf{u})|\mathbf{x}). \tag{1}$$

The features are colour features $\mathbf{z}^C$ and motion features $\mathbf{z}^M$, described in detail in the following sections. Colour and motion features often complement each other in practice. For example, when a hand moves in front of a static skin coloured background, such as the face, motion features can still be discriminative. On the other hand, if the scene contains moving background objects, the colour features are more discriminative. Thus we treat the the features independently at this stage and for both colour and motion features separately sort the subwindows according to their likelihood values $\log p(\mathbf{z}|\mathbf{x})$ normalized by the subwindow size. Local maxima are extracted, while applying non-maximal suppression in order

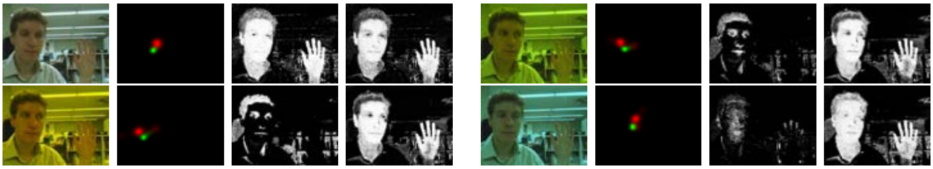


Fig. 2. Colour adaptation based on face detection. Each of the four examples shows the input frame from a sequence with varying colour balance, the smoothed skin colour (green) and background distributions (red) in UV-colour space, probabilities for a static colour model model in normalized UV-space, probabilities for the adaptive model based on face detection.

to obtain a better spatial distribution of the extracted regions. We take the k local maxima of each likelihood map as hypothetical object regions ($k = 3$ in the experiments).

2.1 Colour Model

Skin colour is a useful cue for hand tracking, however one major practical issue is to maintain a robust model under varying conditions: changing illumination, different skin tone of different people and varying background scenes. A popular approach is to work in an intensity-normalized colour space. However, this alone is often not sufficient and a method that is able to initialize and adapt the colour model is required [17, 18, 5, 15]. In this paper the skin colour model is obtained from a frontal face detector, which is run at every kth frame ($k = 30$) and which does not rely on colour information [19]. The assumption is that the skin colour of face and hands have similar distributions, allowing the system to adapt to a specific user as well as the current lighting conditions, even in extreme cases (see figure 2). Colour is represented using a 2D histogram of size 64^2 for vectors $\mathbf{z}^C = [U, V]$, containing the chromatic components of the YUV representation. When a face is detected, the colour values within an ellipse centered and scaled according to a detected location give the distribution $p(\mathbf{z}^C|x^C = \text{skin})$, estimated by histogramming and smoothing the distribution. The background distribution is estimated as a mixture of a uniform distribution and a colour distribution obtained from image regions adjacent to (left, right, and above) the detected face locations.

2.2 Motion Model

Motion is another valuable cue that has been extensively used in HCI applications [18] as it is robust under a wide range of conditions. The motion feature $\mathbf{z}^M$ is computed from the difference image as the L^1-norm of the pixel-wise RGB difference vectors at times t and $t - 1$:

$$\mathbf{z}^M = |R^t - R^{t-1}| + |G^t - G^{t-1}| + |B^t - B^{t-1}| \ . \tag{2}$$

We differentiate between static background and moving objects, not between different foreground objects at this point. The distributions $p(\mathbf{z}^M|x^M = \text{motion})$

and $p(\mathbf{z}^M|x^M = \text{static})$ depend on a number of factors, such as the background scene, the illumination and the motion velocity. We obtain estimates from example sequences of difference images: the distribution for static background is computed from sequences with no moving objects and represents accounts for noise and some lighting variation. The distribution for moving objects is estimated from sequences of a moving hand. The distributions (see figure 1) were found to be reasonably stable across sequences with different backgrounds.

3 Hand Pose Estimation

Given a hypothesized image subregion, the aim is to determine whether the region contains a hand and if so, which pose it is in. The regions are normalized to a size of 40 × 40 pixels and are classified using nearest neighbour classification. We propose two distance measures based on oriented intensity edges and skin colour likelihood, respectively. The next section introduces a distance measure, which is based on oriented edges and avoids the thresholding step of a binary edge detector.

3.1 Oriented Edge Distance

Given a hand image in a normalized window, we compute a descriptor using oriented edge energy [20]. This is based on the response to a pair of even and odd filters, f^e_θ and f^o_θ, respectively, at orientation θ:

$$u^{OE}_\theta = (f^e_\theta * I)^2 + (f^o_\theta * I)^2. \tag{3}$$

Figure 3 shows the filter responses for four example images, showing the advantage of these features in complex scenes over using binary edge maps obtained

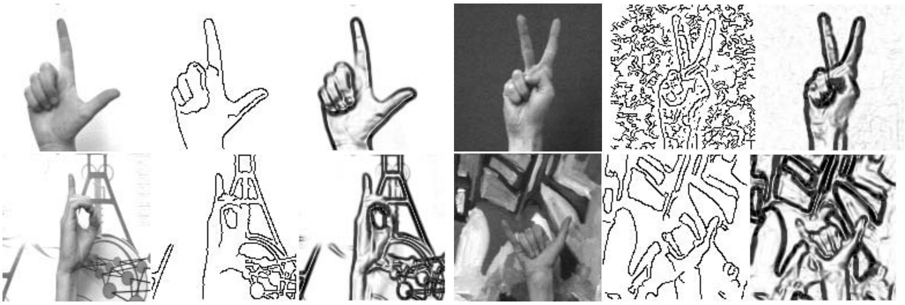


Fig. 3. Oriented edge energy vs. Canny edges. This figure shows the results of the filtering operations on example images from the database of hand images from Triesch and von der Malsburg [7]. For each image triple: **left:** input images, **middle:** Canny edges (constant parameter settings), **right:** oriented edge energy (pixel-wise maximum of responses to four filters with different orientation). The Canny detector often outputs spurious background edges or leads to premature loss of edge information.

with a standard Canny detector, which are widely used to extract geometric information [8, 11]. The value u_θ^{OE} is computed at four discrete orientations for each pixel location, $\theta = \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$, yielding vectors $\{\mathbf{u}_i\}_{i=1,\ldots,4}$, each of which is then normalized and stacked into one vector $\mathbf{u}^{OE}$ [21]. The distance measure between two oriented edge maps $\mathbf{u}^{OE}$ and $\mathbf{v}^{OE}$ is defined as

$$d^{OE}(\mathbf{u}^{OE}, \mathbf{v}^{OE}) = 1 - \frac{1}{4}\sum_{i=1}^{4}\langle \mathbf{u}_i^{OE}, \mathbf{v}_i^{OE}\rangle. \quad (4)$$

The values in each feature vector $\mathbf{u}_i^{OE}$ are then multiplied with the pixel-wise skin colour probability.

3.2 Colour Based Distance

In order to compare templates using colour, the oriented filters are applied to the colour likelihood map. These vectors are also normalized and stacked into one feature vector $\mathbf{u}^C$. The distance measure is defined similar to the previous section as

$$d^{C}(\mathbf{u}^{C}, \mathbf{v}^{C}) = 1 - \frac{1}{4}\sum_{i=1}^{4}\langle \mathbf{u}_i^{C}, \mathbf{v}_i^{C}\rangle. \quad (5)$$

By using the filter responses directly in the distance function, binary thresholding of edges or skin colour at an early stage is avoided.

3.3 Local Template Registration

In order to compute a more exact distance, a continuous image alignment transformation $\mathbf{T}_\alpha$ with parameters α is computed for the best 50 matches. This is necessary to reduce jittered motion, but also to better discriminate between similar poses. A similarity transform is used for this purpose, represented by the 3×3 homogeneous matrix

$$\mathbf{T}_\alpha = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}, \quad (6)$$

where $\alpha = \{s, \mathbf{R}, \mathbf{t}\}$, $s \in \mathbb{R}$ is a scale factor, $\mathbf{R} \in SO_3$ a rotation matrix and $\mathbf{t} \in \mathbb{R}^{2\times 1}$ a translation vector. Similarity transforms are chosen over affine transformations, because shapes under affine transform can look indistinguishable from the hand shape in a different view. The transformation parameters are found by searching over a grid of size 3^4 within the four dimensional parameter space.

3.4 Template Likelihoods

Based on the distances d^{OE} and d^C, a likelihood function is defined for a hand being in a particular pose based on matching to a set of exemplar templates $\{\mathbf{y}_j\}_{j=1,\ldots,N_j}$. The exemplars are then clustered such that each of the original examples is within a certain distance of at least one of the cluster centres

$\{\hat{\mathbf{y}}_k\}_{k=1,\ldots,N_k}$ for both features, edges as well as colour. The feature likelihood of a hand pose is estimated as [22]

$$p(\mathbf{z}|\alpha, k) = \frac{1}{2\pi\sigma^2} \exp -d(\mathbf{z}, \mathbf{T}_\alpha \hat{\mathbf{y}}_k)/2\sigma^2 \tag{7}$$

where $\mathbf{z}$ is the current image observation and $\mathbf{T}_\alpha \mathbf{y}_k$ is the exemplar template $\hat{\mathbf{y}_k}$, transformed by $\mathbf{T}_\alpha$. This term is computed for both oriented edge and colour features, where the parameters are estimated off-line using a set of 500 registered hand images containing variation in illumination and skin coloured background. A pose is detected when the distance value for either oriented edges or colour is below a threshold value, chosen in the experiments as 2σ.

4 Experimental Results

This section shows experimental results the localization task and on two different applications using pose estimation. The first application is a recognition task of ten different hand poses. The second task is a *navigation by pointing* application, where the user can indicate a direction by pointing at the camera. A system overview is given in figure 4. Both experiments are carried out using 320 × 240 images from a single camera directed at the user in an office environment. The skin colour model is initialized using frontal face detection (not shown in the figures). The likelihoods for subwindows is computed at 10 scales.

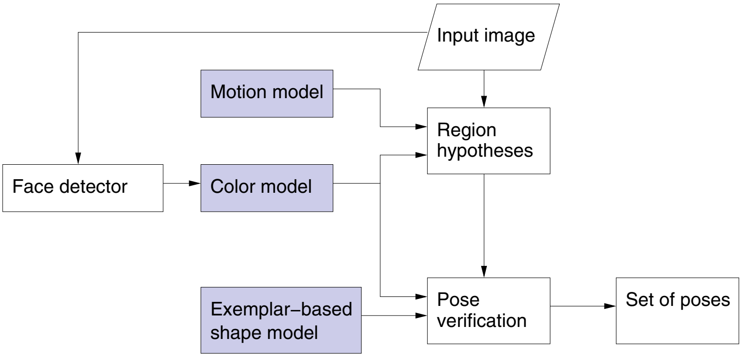


Fig. 4. System overview. In each frame motion and colour features are used to hypothesize image subregions, which are subsequently verified using an exemplar based shape model that includes oriented edge and colour features.

4.1 Pose Recognition

The top of figure 5 shows the ten different poses to be recognized. They are the same poses that have been used in [7]. First, a number of templates are obtained from a sequence taken in front of neutral background. For the recognition stage

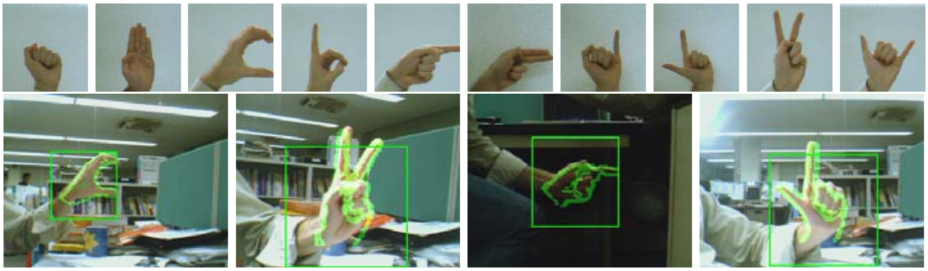


Fig. 5. Hand posture recognition. top: the ten hand postures to be recognized, recorded against a neutral background, **below:** example frames from the sequence with recognition results superimposed. In this sequence the camera is moved for a number of times and there is large variation in lighting conditions.

Table 2. Recognition results for (left) static gesture recognition. This table shows the correctly classified poses on three image sequences (3000 frames each). Each column shows the results of different combinations of localization features (colour/motion) and template features (colour/edges). Different rows show the classification result if the same pose has been classified consistently over a certain number of consecutive frames. **(right) Results for the navigation application.** This table shows the correctly classified poses on four image sequences (3000 frames each) taken of different users.

Consec	col win	mot win	col win	mot win
frames	col tmpl	col tmpl	edge tmpl	edge tmpl
1	0.85	0.33	0.67	0.29
2	0.91	0.50	0.78	0.42
3	0.94	0.63	0.84	0.51
4	0.95	0.73	0.87	0.59
5	0.96	0.79	0.88	0.66

Consec	col win	mot win	col win	mot win
frames	col tmpl	col tmpl	edge tmpl	edge tmpl
1	0.73	0.61	0.57	0.35
2	0.81	0.65	0.69	0.42
3	0.84	0.73	0.74	0.49
4	0.86	0.79	0.80	0.53
5	0.88	0.83	0.84	0.54

40 templates for each pose are used, encoding variation in pose. To increase tolerance to small rotations, for each template a further four templates are generated by rotation around the image centre (-10 to 10 degrees). Three sequences of 3000 frames each were recorded in which a user performed all ten gestures five times. The bottom row of figure 5 shows example frames, demonstrating the variation in scale and lighting conditions. Projected onto each frame is the pose estimate, represented by the closest exemplar template. Table 2 (left) shows the average classification rates in the image sequences according to individual features. Temporal continuity is added by requiring detection of the same pose over a number of consecutive frames. Most errors occur due to confusion of two similar poses (pointing with one or two fingers, respectively). Another error source are highlights on the hand, where pixels have low skin colour probability. Using 1000 templates, the system currently runs at 6 fps on a 3 GHz Pentium IV.

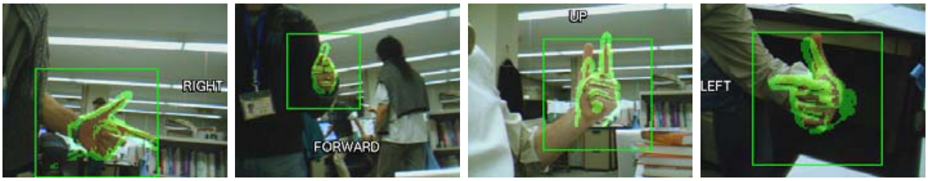


Fig. 6. Navigation by pointing at the camera. This figure shows frames from a test sequence. The overlays represent the estimated pose. Each template is labeled as one of the classes *forward, left, right, up, down, stop*. The examples are from four sequences where three different users perform the gestures.

4.2 Navigation by Pointing

The second application we consider is a navigation by pointing application. In this experiment 300 templates are used for each pose, covering the range of motion of a hand pointing toward the camera. No rotated templates are generated in this case as the orientation is critical for determining the pointing direction. The templates are manually labeled as one of six classes, *forward, left, right, upward, downward, stop*. Four sequences of 3000 frames each were recorded from three different users. Example frames are shown in figure 6. The pose estimate is superimposed, showing that the poses are indeed recognized fairly accurately. Table 2 (right) shows the results on the example sequences. The difficult case here is when the hand is pointing upward, which looks similar to when it points forward with the thumb extended. The system currently runs at 5 fps.

5 Summary and Conclusions

This paper presents a solution to hand pose recognition in cluttered scenes. Initial location hypotheses are obtained using colour and motion, which are verified using normalized template matching with a pre-selected number of templates. The method has been applied to a recognition task with ten poses as well as a navigation by pointing application. A number of techniques have been combined in this system: The colour model is initialized and updated by a frontal face detector. Hand locations and scale are hypothesized efficiently using cumulative likelihood maps, and the hand pose is estimated by normalized template matching. The system lifts several restrictions which are often present in real-time systems: In contrast to other methods, the system in this paper uses neither background subtraction [6] nor does it rely on binary colour segmentation [17, 2, 3] or gesturing at a fixed distance to the camera [23]. Finally, the method is efficient enough to detect the hand in each frame independently. Future extensions of the system include a tracking element that helps to improve the efficiency and accuracy of the pose estimation, as well as an increase in the number of poses.

References

1. Kadir, T., Bowden, R., Ong, E.J., Zisserman, A.: Minimal training, large lexicon, unconstrained sign language recognition. In: Proc. BMVC. (2004)
2. Lockton, R., Fitzgibbon, A.W.: Real-time gesture recognition using deterministic boosting. In: Proc. BMVC. Volume II. (2002) 817–826
3. Tomasi, C., Petrov, S., Sastry, A.K.: 3D tracking = classification + interpolation. In: Proc. 9th ICCV. Volume II. (2003) 1441–1448
4. Bretzner, L., Laptev, I., Lindeberg, T.: Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In: Proc. Face and Gesture. (2002) 423–428
5. Kölsch, M., Turk, M.: Fast 2D hand tracking with flocks of features and multi-cue integration. In: Workshop on Real-Time Vision for HCI. (2004)
6. MacCormick, J., Isard, M.: Partitioned sampling, articulated objects, and interface-quality hand tracking. In: Proc. 6th ECCV. Volume 2. (2000) 3–19
7. Triesch, J., von der Malsburg, C.: Classification of hand postures against complex backgrounds using elastic graph matching. IVC **20** (2002) 937–943
8. Athitsos, V., Sclaroff, S.: Estimating 3D hand pose from a cluttered image. In: Proc. CVPR. Volume II. (2003) 432–439
9. Rehg, J.M., Kanade, T.: Model-based tracking of self-occluding articulated objects. In: Proc. 5th ICCV. (1995) 612–617
10. Shimada, N., Kimura, K., Shirai, Y.: Real-time 3-D hand posture estimation based on 2-D appearance retrieval using monocular camera. In: Proc. Int. WS. RATFG-RTS. (2001) 23–30
11. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Filtering using a tree-based estimator. In: Proc. 9th ICCV. Volume II. (2003) 1063–1070
12. Wu, Y., Lin, J.Y., Huang, T.S.: Capturing natural hand articulation. In: Proc. 8th ICCV. Volume II. (2001) 426–432
13. Wu, Y., Huang, T.S.: View-independent recognition of hand postures. In: Proc. CVPR. Volume II. (2000) 88–94
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
15. Micilotta, A., Bowden, R.: View-based location and tracking of body parts for visual interaction. In: Proc. BMVC. (2004) 849–858
16. Movellan, J.R., Hershey, J., Susskind, J.: Real-time video tracking using convolution HMMs. In: Workshop on Generative Models for Vision. (2004)
17. Argyros, A.A., Lourakis, M.I.A.: Real-time tracking of multiple skin-colored objects with a possibly moving camera. In: ECCV. (2004) 368–379
18. Crowley, J.L., Berard, F.: Multi-modal tracking of faces for video communications. In: Proc. CVPR. (1997) 640–645
19. Viola, P., Jones, M.J.: Robust real-time face detection. IJCV **57** (2004) 137–154
20. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI **26** (2004) 530–549
21. Everingham, M.R., Zisserman, A.: Automated person identification in video. In: 3rd Int. Conf. on Image and Video Retrieval. (2004) 289–298
22. Toyama, K., Blake, A.: Probabilistic tracking with exemplars in a metric space. IJCV **48** (2002) 9–19
23. Von Hardenberg, C., Bérard, F.: Bare-hand human-computer interaction. In: Proc. ACM Workshop on Perceptive User Interfaces. (2001)

Scalable Representation and Learning for 3D Object Recognition Using Shared Feature-Based View Clustering

Sungho Kim and In So Kweon

Dept. of EECS, Korea Advanced Institute of Science and Technology,
373-1 Gusong-Dong, Yuseong-Gu, Daejeon, Korea
{sunghokim, iskweon}@kaist.ac.kr

Abstract. In this paper, we present a new scalable 3D object representation and learning method to recognize many objects. Scalability is one of the important issues in object recognition to reduce memory and recognition time. The key idea of scalable representation is to combine a feature sharing concept with view clustering in part-based object representation (especially a CFCM: common frame constellation model). In this representation scheme, we also propose a fully automatic learning method: appearance-based automatic feature clustering and sequential construction of view-tuned CFCMs from labeled multi-views and multi-objects. We applied this learning scheme to 40 objects with 216 training views. Experimental results show the scalable learning results in almost constant recognition performance relative to the number of objects.

1 Introduction

Object recognition has become mature in terms of identification level with local feature-based approaches. Local features are extracted by the following process: interest point detection [1], region selection [2], and region description [3][4]. Based on these local features, several object recognition methods such as the probabilistic voting method [5], constellation model-based approaches [6], and SVM, Adaboost [7] are introduced. The state-of-the-art methods such as SIFT [3] show very high detection and recognition accuracy in general environments. However, as the number of objects increases, the issue of scalability becomes more important. Conventional object representations require linear memory and recognition time proportionate to the number of objects. This problem can be more severe if the objects are 3D. Storing all the multiple views of 3D objects is almost impractical.

Recently, some feasible approaches have been proposed to alleviate the scalability problem. Torralba et al. [8] modified the Adaboost to recognize multiclass object using a feature-sharing concept. They demonstrated that shared features outperform independently learned features. Murphy-Chutorian and Triesch adapted feature clustering to solve the problem [9]. This method recognizes objects by nearest voting using the clustered-feature database. Lowe proposed a local feature-based view-clustering scheme to represent multiple views of 3D

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 561–570, 2006.

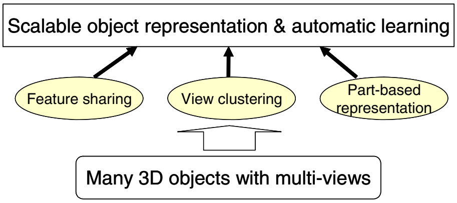

Scalable object representation & automatic learning
Feature sharing
View clustering
Part-based representation
Many 3D objects with multi-views

Fig. 1. Key idea of scalable object representation: We apply the feature sharing and view clustering to multiple views of 3D object for scalable object representation

objects [10]. But, these approaches are partial works to minimize the scalability problem in terms of feature level and multiple view level.

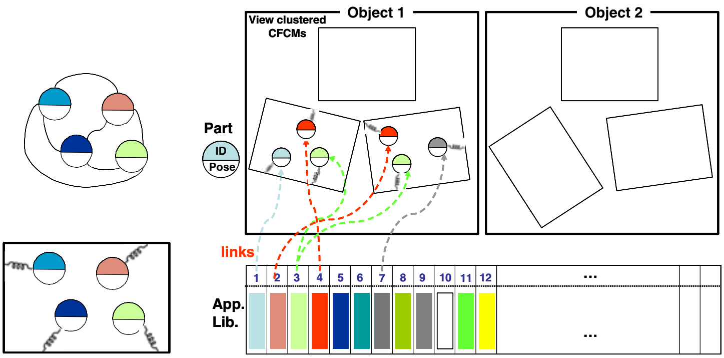
How can we reduce the DB size from many objects and views without degrading recognition performance? In this paper, we present a new object representation and learning method by combining a feature-sharing concept [9] and a view-clustering concept [10] in part-based object representation [6] as shown in Fig. 1. In Section 2, we introduce a scalable 3D object representation scheme. Sections 3 and 4 explain a proposed learning method for the representation. Section 5 details a recognition method for the validation. In Section 6, we demonstrate the scalability of the proposed method and conclude in Section 7.

2 Scalable 3D Object Representation

As we discussed, simply storing all possible views of many 3D objects requires huge memory and recognition time. The main cause is originated from the redundancy in DB generation. We have to remove the redundancies effectively to get minimal DB construction. In advance, we adapt a part-based object representation, specifically a common-frame constellation model (CFCM) [6] instead of a holistic appearance representation. The CFCM representation scheme provides useful advantages in terms of computation and redundancy.

***Computational efficiency*:** An object can be represented as a set of visual parts. The well-known mathematical model is a fully parameterized constellation model as in Fig. 2 (a) (top) [11]. The circle means an object part which contains appearance information and part pose. If each part is x_i and the number of part N, then it can be modeled as full covariance-based joint pdf, $p(x_1, x_2, \ldots, x_N)$. The DOF (degree of freedom) of required parameters $O(N^2)$. However, if we fix the object ID and view point, each part can share the viewing parameters, $\theta = [objectID, pose]$ as Fig. 2(a) (bottom) [6]. Then the mathematical representation can be reduced to the product form conditioned on an object parameter like $\Pi_{i=1}^{N} p(x_i|\theta)$. In this scheme, the order is reduced to $O(N)$ which is useful during object recognition. We refer to this part-based object representation as CFCM since each part shares object parameters $(objectID, pose)$.

***Easy redundancy removal*:** In a CFCM, we can find the source of redundancies easily. One source of the object parts and the other is the object parameters



(a) (b)

Fig. 2. (a) (top) Fully connected constellation model: It can model objects which contain up to 5 7 parts. (bottom) Common-frame constellation model: It can model objects which have hundreds of parts. (b) Any 3D objects can be rerepsented by shared part-based CFCMs which are view clustered.

of object ID and view point. Since training images are composed of many multiple views of 3D objects, there exist redundant parts and views. We can reduce the redundancies by applying a clustering concept to both parts and views.

Based on these motivations, the proposed scalable object representation framework is shown in Fig. 2(b). The bottom table is the feature (appearance of part) library. Each feature represents an appearance vector which is obtained by vector clustering. The appearance feature of an individual part can be anything such as a SIFT descriptor [3], PCA [12], or moments [13]. A 3D object is represented as a set of view-clustered CFCMs. Each CFCM contains object parts which have part pose and the link indices to part libraries (appearance). The part pose represents part size, part orientation, and position in a CFCM. These kinds of information are available in [1][3]. Likewise, each element in the library contains all the links to the parts in the CFCMs. We can use this fact to generate hypotheses during object recognition. The next two sections explain the details of learning by feature and by view clustering respectively.

3 Visual Feature and Clustering

3.1 Generalized Robust Invariant Feature

We detect visual parts based on object structures. First, high-curvature points and radial symmetry centers are extracted using the Harris corner and DoG (difference of Gaussian) methods respectively. Second, part size is determined at the local maxima of convexity where DoG is compared in scale space (see Fig. 3). This method can extract complementary object parts. Dominant orientation of visual part is calculated using a weighted steerable filter. Finally, the detected convex part is encoded using a set of localized histograms (a total of 21) of edge

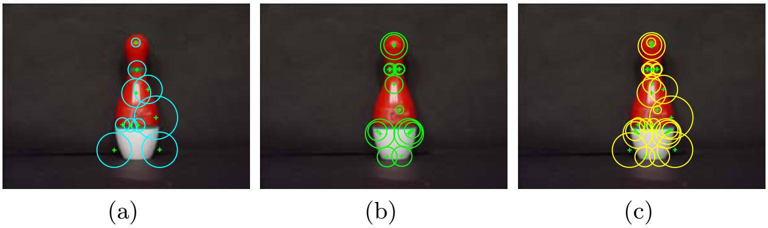


Fig. 3. We can detect structure-based object parts: (a) radial symmetry part (b) corner-like part (c) complementary visual part detector (proposed) [14]

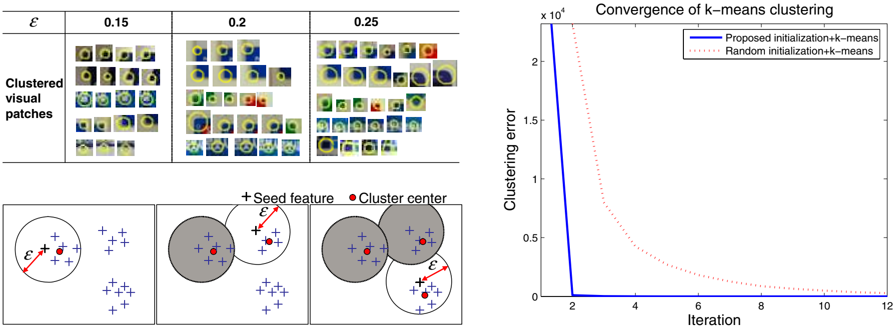
orientation (4 bins), edge density (1 bin), and hue (4 bins). This is a generalized form of contextual descriptor [3][4]. The feature dimension is 189 (21*(4+1+4)). More details is explained in [14]. We call the feature G-RIF for its properties. We will use the term G-RIF throughout this paper.

3.2 Automatic Feature Clustering

A feature library or code book can be generated by feature clustering from training features. There are several clustering methods, such as k-means algorithm, vector quantization [15]. These methods are based on iterative optimization starting from random cluster centers with a predetermined number of clusters. In our database, the dimension of a feature is over hundred (189) and the size of a feature is more than several hundred thousand. In this case, the conventional energy minimization-based approach is impractical due to the convergence time. The main problems of k-means algorithm for huge data are:

- How to set the cluster size.
- How to set the initial cluster centers.
- How to effectively compare distances between data and cluster centers.

We propose a simple and practical clustering algorithm suitable for high dimensional visual features. We solve the above problems by utilizing the properties of part structures, and a nearest-neighbor search using a k-d tree [16]. As we can see in Fig. 4(a) (top), we can cluster visually similar parts using only the distance threshold (ε) between normalized feature vectors. As the threshold becomes larger, roughly similar structures are clustered. In part-based object recognition, part structures have very important roles. So, first we find rough structure centers by sequentially performing the ε-nearest neighbor search as in Fig. 4(a) (bottom). The clustered features are removed in search space. Then, cluster centers are optimized using k-means clustering. This process corrects the features on the cluster boundaries. By merging the ε-nearest neighbor search, k-d tree-based distance calculation, and k-means algorithm, we can solve the above three problems simultaneously. Fig. 4(b) shows the convergence rate of clustering with the proposed initialization and random samples in k-means clustering. The proposed automatic clustering is almost converged within two iterations due to the good initial estimation of cluster centers.

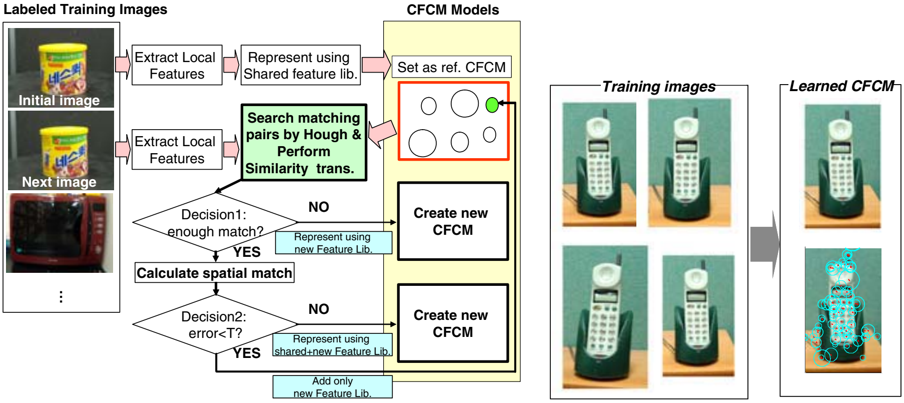


(a) (b)

Fig. 4. (a) ε-NN search results from training parts (top) automatic sequential clustering proces (bottom), (b) Convergence comparisons between the proposed automatic clustering and conventional k-means algorithm

4 Sequential Construction of Scalable Object Model

As we said, we represent a 3D object by a set of view-tuned CFCMs. Visual parts in a CFCM are conditioned on the view-tuned parameters. The term *view-tuned* means view clustering in a similarity transform space. Fig. 5(a) shows the overall object learning structure. Given labeled multi-views and multi-object images, we have to find view-tuned CFCMs. In a CFCM, each part is represented in terms of pose and the appearance index to the shared feature libraries learned in the



(a) (b)

Fig. 5. (a) Object learning by the sequential view clustering: Given appearance library, image features are extracted from each training image. Then proper clustering action (Case I, II, III) is selected based on the the reuslt of core functional blocks in Fig. 6. (b) An example of view clustering: The 4 training images are represented by a view-tuned CFCM.

previous section. The learning is conducted sequentially. We set the first image as a reference CFCM. A CFCM contains object, view ID, and parts (pose, appearance index per part). The pose of a part is obtained directly from the feature detector and the part appearance is represented using the index of clustered appearance library. From the next image, we extract local features. Matching pairs are searched for between the input feature and the CFCM in the DB using a Hough transform in pose space (CFCM ID, scale, orientation space, see Fig. 6). Finally, new CFCMs are constructed according to the following three cases.

CASE 1: If there are few matching pairs ($T1 < 5$), then generate a new CFCM represented in terms of new appearance libraries.
CASE 2: If there are enough matching pairs but the spatial average matching error by similarity transformation is below a predefined threshold (T2: 4 pixels), then create a new CFCM with a shared-feature and with new feature libraries.
CASE 3: Finally, if the matching pairs are matched almost correctly, then add distinctive new features to the model CFCM. Distinctive features can help the discrimination during inference. We define that a new input feature (f_{New}) is distinctive if it is close to the shared feature library (f_{Lib}) and the sharing number of the library feature ($N_{Shared}^{f_{Lib}}$) is as low as possible as expressed in eq. 1.

$$DM(f_{New}\text{‘}\, f_{Lib}) = \frac{exp(-N_{Shared}^{f_{Lib}})}{dist(f_{New}\text{‘}\, f_{Lib})} \tag{1}$$

Fig. 6 shows the core functional blocks of the sequential CFCM construction method by view clustering. From an input image, we extract visual features as Fig. 6(a) then find a candidate CFCM (Fig. 6(c)) by the Hough transform in pose space (Fig. 6(b)). Finally, we check the similarity transform error between the two images (Fig. 6(d)). Fig. 5(b) shows an example of sequential CFCM construction results from multiple object views. We obtain one view-tuned CFCM image from 4 training images. The proposed CFCM construction method can extract distinguishable multiple views for 3D objects in similarity transform space (affine transformation is not suitable for 3D objects since the feature detector is robust up to similarity transform).

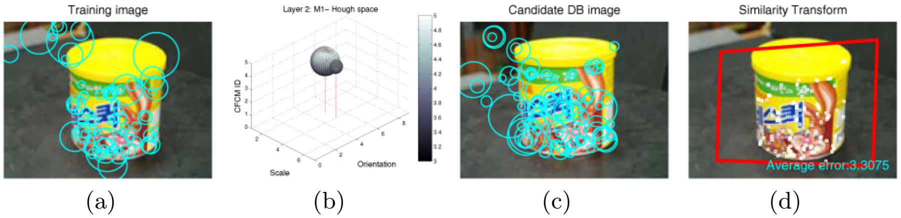


Fig. 6. Core functional blocks of CFCM construction: From a training image with detected features (a), candidate CFCM (c) is found by Hough transform in pose space (b). View similarity transform error is used to determine the view clustering (d).

5 Multi-object Recognition

How can we fully utilize the shared feature-based view clustering method in object recognition? Basically, we modify the well-known hypothesis and verification framework for recognizing multiple objects with the proposed object representation scheme.

If S represents a set of scene features, D represents a set of database entries (shared feature lib. + CFCMs), and H is hypothesized CFCMs which describe the scene best, then the objects recognition problem can be formulated as a mixture form (the 1st line in eq. 2: assume multiple objects in a scene). π_m is the mixture weight of object m which is estimated on-line by a set of CFCMs belonging to m. $\hat{h}_m$ is the optimal transformed CFCM for object m. If we assume uniform priors, the equation can be reduced to the 2nd line in eq. 2. We select the best hypothesis ($\hat{h}_m$) which has the maximal conditional probability ($p_m(S_m|h_m^{(i)}, D)$).

$$\begin{aligned} p(H|S, D) &= \textstyle\sum_{m=1}^{M} \pi_m p_m(\hat{h}_m|S_m, D) \\ &\propto \textstyle\sum_{m=1}^{M} \pi_m p_m(S_m|\hat{h}_m, D) \end{aligned} \tag{2}$$

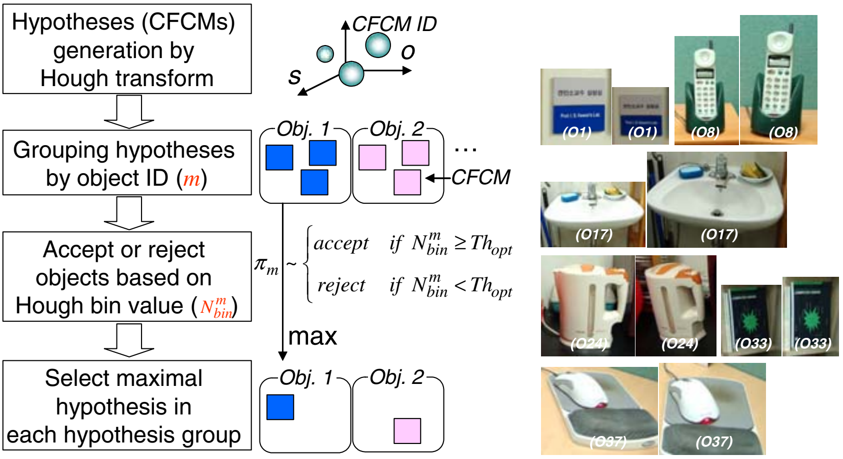
where $p_m(S_m|\hat{h}_m, D) = \arg\max_{i \in I_m}\{p_m(S_m|h_m^{(i)}, D)\}, \sum_{m=1}^{M} \pi_m = 1$. We can model $p_m(S_m|h_m^{(i)}, D)$ by a Gaussian noise model of appearance and pose using eq. 3. We assume that the appearance and pose of each part is independent. In addition, since features in a CFCM are conditioned on a common-frame, they can be handled independently. $\mathbf{y}_{app}$ is the shared feature closest to scene feature $\mathbf{x}_{app}$. $\mathbf{y}_{loc}$ is the position of a part hypothesized by $h_m^{(i)}$. σ_{app}, σ_{loc} are estimated from training data during sequential CFCM constrcution as Fig. 5(a).

$$p_m(S_m|h_m^{(i)}, D) = \prod_{\mathbf{x} \in S_m} p_{app}(\mathbf{x}|h_m^{(i)}, D) \cdot p_{pose}(\mathbf{x}|h_m^{(i)}, D) \tag{3}$$

where $p_{app}(\mathbf{x}|h_m^{(i)}, D) \propto exp(-\|\mathbf{x}_{app} - \mathbf{y}_{app}\|^2/\sigma_{app}^2)$, $p_{pose}(\mathbf{x}|h_m^{(i)}, D) \propto exp(-\|\mathbf{x}_{loc} - \mathbf{y}_{loc}\|^2/\sigma_{loc}^2)$. Fig. 7(a) summarizes the object recognition procedures graphically. We can get all possible matching pairs by an NN (nearest neighbor) search in the feature library. Hypotheses are generated by Hough transform in the CFCM ID, scale (11 bins), orientation (8 bins) space [3], and grouped by object ID. Then we accept or reject the hypothesized object based on the bin size with an optimal threshold [9]. Finally, we select the optimal hypotheses using equation (3) which can best be matched to object features in a scene.

6 Experimental Results

We prepared 40 3D objects to test the scalability of the proposed method. The object are segmented and labeled as shown in Fig. 7(b). The total number of training views is 216. We use 90 test scenes where each scene contains 0∼6 objects (total: 247 objects) not used in the learning process.


Hypotheses (CFCMs) generation by Hough transform
Grouping hypotheses by object ID (m)
Accept or reject objects based on Hough bin value (N_{bin}^{m})
Select maximal hypothesis in each hypothesis group
CFCM ID
o
s
Obj. 1
Obj. 2
…
CFCM
$\pi_m \sim \begin{cases} accept & if\ N_{bin}^{m} \geq Th_{opt} \\ reject & if\ N_{bin}^{m} < Th_{opt} \end{cases}$
max
Obj. 1
Obj. 2
(O1)
(O1)
(O8)
(O8)
(O17)
(O17)
(O24)
(O24)
(O33)
(O33)
(O37)
(O37)

(a) (b)

Fig. 7. (a) Multiple object recognition is conducted by hypothesis and test (verification). Hypotheses of all possible CFCMs are generated from Hough transform of matching pairs. CFCMs are grouped then maximal CFCMs are selected for each group. (b) Partial examples of labeled multi-view/multi-object images for training.

Table 1. The size of clustered features and CFCMs is reduced as the thresholds (ε, $T2$ respectively) increase

ε	0	0.1	0.2	0.3	0.4	0.5
No. of shared feature	35,027	32,627	24,957	18,439	14,061	8,472
$T2$	0	1	2	4	10	20
No. of view-tuned CFCM	216	179	147	120	100	97

Given the huge training data set, we first extract all visual features using G-RIF [14]. Then we apply automatic clustering as Fig. 4(a). The clustered features are stored in a k-d tree structure to reduce the search time by $O(\log(N))$. Based on these clustered features, we sequentially construct view-tuned CFCMs using the learning method shown in Fig. 5(a). The size of the view-tuned CFCM is determined by the threshold ($T2$) of similarity transform error. Table 1 summarizes these learning results by changing two parameters. We can reduce the number of features by feature sharing and also reduce the size of the CFCMs by view clustering.

Next, we evaluated the recognition performance for the various learning data shown in Table 1. We decide that a recognition result is successful if both object ID and pose are correct by human eye. Fig. 8 shows the results. Fig. 8(a) is obtained by fixing the size CFCM at 100 ($T2$=4). As the feature size increases, the system shows higher recognition rate. However, the recognition is almost

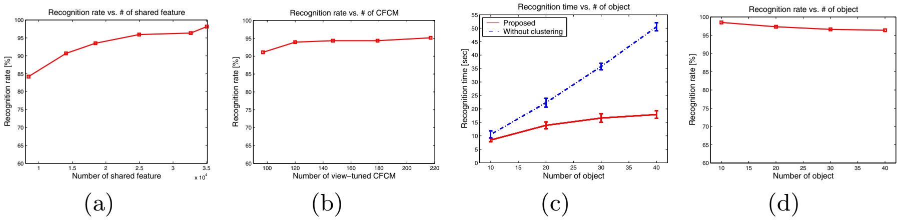


Fig. 8. Evaluation of object recognition according to database size: (a) number of shared feature, (b) number of view-tuned CFCMs, (c) Recognition time vs. number of objects (d) recognition rate vs. number of objects

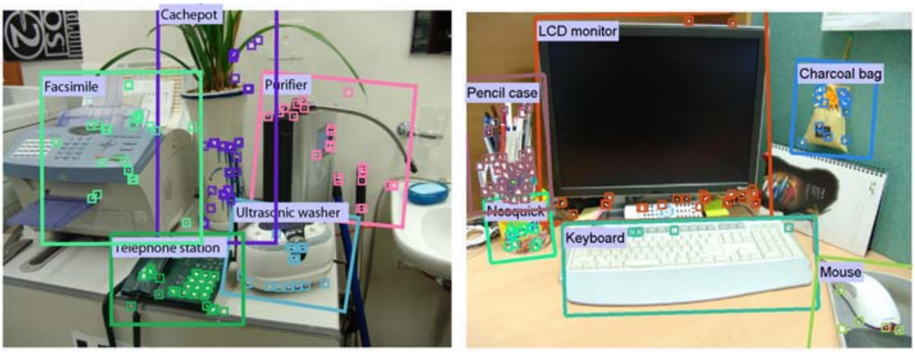


Fig. 9. Object recognition results by hypothesis and verification scheme

converged at the size of feature 24,957 (ε =0.2, 95%). Likewise, we can get the recognition performance according to the size of a view-tuned CFCM (we fix the number of feature to 24,957) as Fig. 8(b). We can get very high accuracy with only 120 CFCMs ($T2$=4, 95%).

Finally, we checked the recognition time and recognition rate according to the number of objects. Note that the recognition time is log-linear to the number of object as Fig. 8(c). Furthermore, the recognition rate is almost constant to the number of object as Fig. 8(d). From these experiments, we can determine if the proposed object representation scheme is scalable. If we set ε to 0.2, $T2$ to 4, the overall recognition rate is 95.8% with false alarm rate 2.43% (this is, rate of incorrect poses). Fig. 9 shows examples of multiple object recognition results by selecting maximal CFCMs in multimodal probability in eq. 2.

7 Conclusions

In this paper, we focus on scalable 3D object representation and its learning by combining feature sharing and view clustering in part-based recognition. Visual structure-based automatic clustering is especially useful to feature sharing and sequential construction of CFCMs that can learn any new incoming objects of practical importance. We experimentally validate that the shared feature-based view clustering scheme can effectively represent 3D objects and is scalable to the number of objects. We recognize multiple objects by a hypothesis and verification method in identification level. We will next investigate how to upgrade the scalable object representation and learning scheme to the categorization level.

Acknowledgements

This research has been supported by the KOTEF grant and the Korean Ministry of Science and Technology for National Research Laboratory Program (M1-0302-00-0064), Korea.

References

1. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. IJCV **37** (2000) 151–172
2. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. IJCV **60** (2004) 63–86
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and bject recognition using shape contexts. PAMI **24** (2002) 509–522
5. Schmid, C.: A structured probabilistic model for recognition. In: CVPR'99. (1999)
6. Moreels, P., Maire, M., Perona, P.: Recognition by probabilistic hypothesis construction. In: ECCV'04. (2004) 55–68
7. Wallraven, C., Caputo, B., Graf, A.B.A.: Recognition with local features: the kernel recipe. In: ICCV'03. (2003) 257–264
8. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing features: efficient boosting procedures for multiclass object detection. In: CVPR'04, Washington, DC (2004) 762–769
9. Murphy-Chutorian, E., Triesch, J.: Shared features for scalable appearance-based object recognition. In: WACV'05. (2005) 16–21
10. Lowe, D.G.: Local feature view clustering for 3d object recognition. In: CVPR'01. (2001) 682–688
11. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR'03. Volume II. (2003) 264–271
12. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: CVPR'04. (2004) 506–513
13. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI **27** (2005) 1615–1630
14. Kim, S., Kweon, I.S.: Biologically motivated perceptual feature: Generalized-robust invariant feature. In: ACCV'06. (2006) To appear
15. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication (2000)
16. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM **45** (1998) 891–923

Video Scene Interpretation Using Perceptual Prominence and Mise-en-scène Features

Gaurav Harit and Santanu Chaudhury

Indian Institute of Technology, Hauz Khas, New Delhi, India
{gharit, santanuc}@ee.iitd.ac.in

Abstract. We propose an empirical computational model for generating an interpretation of a video shot based on our proposed principle of perceptual prominence. The principle of perceptual prominence captures the key aspects of mise-en-scène required for interpreting a video scene. We present a novel approach for applying perceptual grouping principles to the spatio-temporal domain of video. Our spatio-temporal perceptual grouping scheme, applied on blob tracks, makes use of a specified spatio-temporal coherence model. A high level semantic interpretation of scenes is done using the mise-en-scène features and the perceptual prominence computed for the perceptual clusters.

1 Introduction

Our understanding of the scene draws mainly from the characteristics and the context of the scene components we focus on. What we focus onto depends on the mise-en-scène and also our world knowledge. The perceptive and the cognitive phenomena work together to direct our attention. Computer vision techniques can be exploited to identify the perceptible attributes. Our world knowledge and expectations, together with the temporal behavior analysis of the *perceptible* attributes taking the context into account, constitute as inputs for computing the *perceptual* attributes of the scene components.

In this work we propose the concept of *perceptual prominence* for the subjects in the scene. Prominence can have different interpretations, which may lead to different specifications of cognitive interest in the perceptual attributes of the subjects. The specifications can be parametrized into different *prominence models*. A prominence model specifies a set of perceptual attributes, and a prominence function which uses them to compute a measure signifying the prominence.

Different subjects may turn out to be prominent when using different prominence models. Video shot interpretation follows from the analysis of the prominent subjects in the scene. We do not attempt to model the psychophysical sensory mechanisms of human visual system, as is done in Visual attention [1]. We deal with prominence arising as a result of "perceiving", ie, the cognitive interest coming up as a result of understanding and awareness of the complete visualization space (2D + time).

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 571–580, 2006.

Several approaches have made use of perceptual organization to identify meaningful structures in images and videos. [2] have used perceptual grouping to estimate a dense velocity field given a pair of frames. [3] have presented a perceptual grouping based algorithm for segmenting motion trajectories. [4] have applied the principle of perceptual organization to track geometrical structures like rectangles, quadrilaterals, ellipses, circles, ribbons, triangles, etc. [5] have worked on perceptual grouping in spatio-temporal domain using motion cues. Pascal et al. [6] have formulated a perceptual organization approach based on Dempster Shafer Theory.

Prior attempts to quantify the saliency of perceptual structures have made use of structural information, local characteristics, and probability of occurrence of the structure [7], [8], [9], [10]. We have extended the concept of saliency to be evaluated at the cognitive level, and term it as perceptual prominence. We make use of context and mise-en-scène to compute the perceptual prominence.

We identify the subjects using perceptual grouping principles. Perceptual organization deals with identifying inherent organizations in primitives which can lead to more meaningful structures. We develop here an empirical computational model for classifying a video scene into two broad categories: *subject-centric* scenes, that have one or few prominent subjects, and, *frame-centric* scenes, where none of the subjects can be attributed a high prominence and hence the entire frame is a subject of interest.

The contributions of our work lie in the use of a novel clustering methodology for identifying space time regions homogeneous in color, a novel perceptual grouping algorithm which identifies the foreground clusters (subjects) by grouping the blob tracks of homogeneous color regions, use of a novel principle of perceptual prominence to generate a taxonomy of interpretation of video scenes, and the use of mise-en-scène features for scene classification.

2 Space-Time Homogeneous Color Regions by Clustering

In this section we provide an overview of our clustering methodology to identify tracks of homogeneous color regions modeled as 2-D blobs. The 6-D feature space of video data (3 for color, 2 for position and 1 for time coordinates) is a semantically heterogeneous feature space because color, position and time are features with different semantics. Our scheme uses a decoupled clustering along different feature dimensions. The set of samples within a cluster are further organized into subclusters in the subsequent level of the hierarchy. The hierarchical sequence of clusters can be represented in the form of a tree which we refer to as the *Decoupled Semantics Clustering tree* (DSCT). Selection of the clustering scheme (and thus the clustering model) to be used for a given hierarchy level is done by taking into account the nature of data distribution along the feature set chosen for partitioning at that level. At the first level, the feature space is partitioned along time to give video stacks, each of size 10 frames. For the second level, we do a color clustering of the 3-D color (in LUV space) data for the pixels in the video stack. The color clustering is done using hierarchical mean shift [11]. The color model comprises of color modes selected from the mean shift dendrogram. Selection of the color model is done such that the color distance between the color modes is less than a threshold τ_c (chosen 12). At the third level, each color cluster is partitioned along time to obtain a projection of the color cluster on the frames, yielding a set of pixels depicting regions homogeneous in that color for each frame of stack. At the fourth level, we apply GMM clustering on the (x,y) position features of these pixels to model them as 2-D blobs. As a final step, we generate tracks of these 2-D blobs using a criteria of smooth motion of the blobs from one frame to another. The blobs are tracked within the stacks and then across the stacks using a criteria of smooth motion and color similarity. The blob tracks are also linked across occlusions by formulating a track similarity measure.

3 Perceptual Grouping in Spatio-temporal Domain

The space time homogeneous color regions (blob tracks) need to be organized into meaningful clusters using a grouping process. For the spatio-temporal domain, the grouping criteria are formulated using attributes that characterize the spatial coherence as well as the temporal characteristics so that the grouping persists in time. In our formalism, the Gestalt principle of common fate [12] is functionally modeled as the temporal consistency of attributes of a spatial organization of patterns. A grouping is valid in the spatial domain, if it follows Gestalt principles such as connectedness, proximity and similarity of motion. We quantify the grouping saliency measure of a cluster as the grouping probability of the cluster. Our computational model for evaluating a spatio-temporal grouping is a belief network which provides the grouping probability of a cluster c_j. The belief network is shown in Fig 1 where the node S takes on two values [*groupingIsSalient, groupingIsNotSalient*]. The grouping probability P(c_j) computed at node S is evidenced on a set of Gestalt criteria (the nodes A, B, C, D, E) for spatio-temporal grouping.

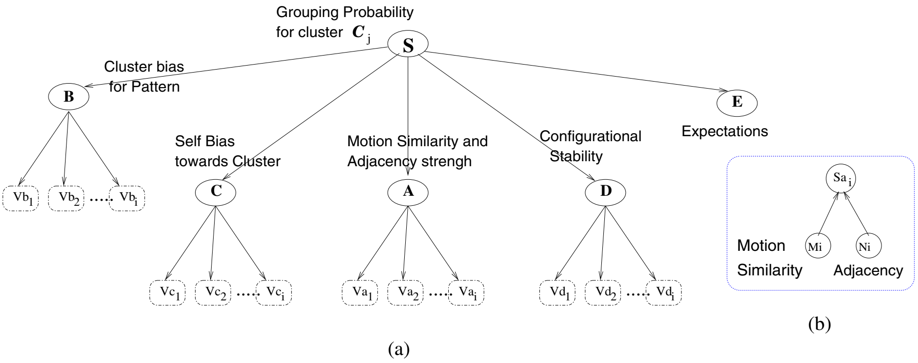


Fig. 1. Belief Network for Spatio Temporal Grouping

For computing the virtual evidence at each of the nodes (A,B,C,D,E) we analyze the attributes of associations between patterns to evaluate the specific Gestalt associations which lead to a salient grouping. Each pattern in a putative cluster contributes an evidence for the saliency of the grouping. An evidence is computed by evaluating a specified Gestalt association of a pattern p_i *to the rest of the cluster* c_j. If a cluster c_j has a pattern that does not form a valid association (say association type g) with the remaining patterns of the cluster, then the computed evidence by evaluating that association would lead to a low probability of grouping $P_g(c_j)$. The pattern with the longest lifespan in the cluster is referred to as the *generator* of the cluster. We take the generator node as the representative pattern of a cluster. We approximate the spatial association measure of pattern p_i with cluster c_j using the spatial association between p_i and the generator of c_j. We next describe the computation of evidences (at the leaf nodes Vm_i, Vn_i, Vb_i, etc) for each of the 5 different grouping criteria formulated as associations between a pattern and a cluster.

(A) *Motion Similarity and Adjacency:* This association plays the important role of grouping the foreground blobs having a distinct motion relative to the background blobs. We compute the frame to frame motion vector for a blob track as the displacement between the blob centroids. The evidence is proportional to the average difference in the motion vectors of the two patterns computed over all the frames in which the patterns exist simultaneously. Adjacency evidence is formulated as proportional to the number of frames for which the pattern has an overlapping boundary with the cluster.

(B) *Cluster bias for a Pattern:* The cluster bias signifies the affinity of the cluster towards the pattern. A pattern may be important for the cluster if it could facilitate the adjacency of other patterns towards the cluster. For example, removing a pattern p_i may cause a set of patterns q to get disconnected from the cluster. The cluster bias for p_i is formulated as proportional to $\sum_{\forall q_k \in q} d_k$, where d_k is the period for which the pattern $q_k \in q$ gets disconnected from the cluster

(C) *Self bias of a Pattern:* It signifies the pattern's affinity towards a cluster. A pattern will have a self bias to a cluster if it happens to share an adjacency to the cluster for a temporal period which is a large fraction of its own lifespan. These are the patterns which remain mostly occluded during the cluster lifespan and appear for only short durations. The self bias is proportional to the fraction of the pattern's lifespan relative to the cluster lifespan.

(D) *Configuration Stability:* There are situations when it is desirable that the relative geometrical configuration of the patterns in a cluster be stable for the grouping to be valid. Our formulation for configuration stability quantifies the relative change in the configuration of the pattern with respect to other patterns in the cluster.

(E) *Expectations:* Recognition of a part of an object is a cue for the presence of other parts, which can be identified and grouped to form the complete object. For example, in scenes where the subjects are stationary, we use a face detector to identify the face regions. A face region is then grouped with other blobs below it to delineate the human body.

4 The Perceptual Grouping Algorithm for Cluster Identification

In this section we discuss our algorithm for identifying clusters from a given set of spatio-temporal patterns (blob tracks). The grouping saliency $\mathcal{S}$ of a cluster is a measure of goodness of the grouping. We formulate the grouping saliency measure of a cluster as the grouping probability for the cluster, computed using the belief network of Fig 1. The grouping saliency of a cluster c_j denoted as $\mathcal{S}_{c_j}$ is computed as: $\mathcal{S}_{c_j} = \mathrm{P}(c_j)$, where $\mathrm{P}(c_j)$ is the grouping probability of the cluster c_j. If C be the set of all perceptual clusters in the scene, then the grouping saliency for the entire scene is the sum $\mathcal{S}_{scene} = \sum_{\forall c_j \in C} \mathcal{S}_{c_j}$. The spatio-temporal grouping problem is to identify the set C of perceptual clusters in the scene so as to maximize $\mathcal{S}_{scene}$ such that the final set of clusters have

$$\mathrm{P}(c_i) \geq 0.5, \quad \mathrm{P}(c_i \cup c_j) < 0.5, \quad \text{and} \quad c_i \cap c_j = \phi \qquad \forall \; c_i, c_j \in C, \quad i \neq j \quad (1)$$

We have thus formulated the perceptual grouping problem as an optimization problem. We outline our algorithm which maximizes $\mathcal{S}_{scene}$ to a local maximum, while satisfying constraint 1.

Step 1. Instantiate a new cluster if there are patterns without any cluster label, ie which do not belong to any of the clusters. From amongst the unlabeled patterns, pick up the one that has the maximum lifespan. This pattern is called as the generator pattern for the new cluster. Put the newly instantiated cluster, which right now consists of only the generator pattern, into a queue *clusterQ*. If all the patterns have cluster labels and the *clusterQ* is empty, then exit.

Step 2. Take the cluster at the front of the *clusterQ*. Call this cluster c_f.

Step 3. Consider a pattern p_i of the cluster c_f. Compute the grouping saliency of this pattern with every cluster. From amongst the clusters that form a salient grouping with this pattern, choose the one for which the increase in the scene's grouping saliency is maximum. If the chosen cluster is different from the existing cluster label of p_i, then relabel p_i. As a result of relabeling, some patterns may leave membership of c_f and some patterns may become new members of c_f. If any pattern changes its cluster label from k to j, then insert the clusters c_k and c_j into the *clusterQ* if they are not already present in the queue.

Step 4. If any pattern changes its label as a result of step 3, then go to step 2. If none of the patterns changes its label, remove the front element (c_f) of the queue and go to step 1.

The iterative process terminates when the pattern labels have stabilized. It attempts to hypothesize new clusters and organizes them such that $\mathcal{S}_{scene}$ reaches a local maximum. Convergence to a local maximum is guaranteed since we ensure that at every step of the algorithm, $\triangle\mathcal{S}_{scene} \geq 0$. The foreground and the background blobs get grouped into separate clusters.

5 Perceptual Prominence

We first give the formal definition of Perceptual Prominence and then provide the illustrations. Let $S = \{p_1, p_2, .., p_n\}$ be a set of elementary patterns. Let $C = \{c_1, c_2, .., c_m\}$, such that $c_1 \cup c_2 \cup, ..., \cup c_m \subseteq S$, where $c_i \cap c_j = \phi$, $i \neq j$, be a set of perceptual clusters identified using the spatio-temporal grouping algorithm. The members of each cluster c_i share some set of common characteristics c_i^*. This set of common characteristics *may* be defined distinctly for different clusters. Let ψ denote the *perceptual attributes* of a cluster. These attributes are common to all clusters. Denoting ψ_i as the value of ψ for c_i, let $\Psi = \{\psi_1, \psi_2, \psi_3, ..., \psi_k\}$ be the set of perceptual attributes of all the clusters in C. *The perceptual prominence $\mathcal{P}(c_i)$ for a cluster c_i is defined as its contextual perceivability under some interpretation $\mathcal{I}$. The interpretation $\mathcal{I}$ specifies a vector of perceptual attributes ψ and a methodology which gives a prominence measure for the cluster c_i in the context Ψ.*

The perceptual attributes of a cluster characterize the contextual behavior and appearance of the cluster. A contextual attribute captures the distinguishing properties of the cluster with respect to the neighboring region or the other clusters. Not all attributes need to be contextual. When all the perceptual attributes are computed without taking the context into account, the prominence measure is called as the *self prominence*. Our methodology for computing the prominence measure is a belief network which models the prominence of a cluster as a proposition (a node in the network) and the observable nodes of the belief network are the perceptual attributes which constitute the evidence for the prominence. The prominence measure is formulated as the probability computed for the prominence. Our belief network for modeling the prominence is shown in Fig 2. The conditional probability tables used for the belief network dictate how the various evidences influence the prominence measure.

Prominence can be characterized in several ways. For example, in a subject oriented scene, the leading subject generally lasts for a long duration and hence

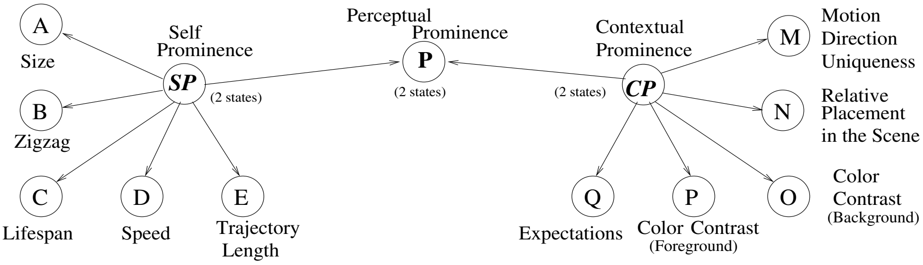


Fig. 2. Belief Network for a Prominence Model

can be identified as prominent. In surveillance scenes, subjects which appear for a short span of time, or subjects which move in a direction different from other subjects are regarded as prominent (deserving attention). In many scenes, a subject occupying a position so as to create an imbalance in the arrangement of the scene components, turns out to be prominent. The perceptual attributes leading to prominence are formulated as virtual evidences. Each virtual evidence provides beliefs in favor and against the proposition of prominence. Different interpretations of prominence can be encoded into different prominence models.

6 Scene Interpretation

Scene interpretation involves the recognition and analysis of scene components which comprise of both the background and the foreground subjects. A specific type of a scene dictates a specific composition and behavior model for its constituent subjects. The perceptual attributes for the subjects show adherence to the scene-type-specific behavior model (henceforth referred to as the the scene model). Our model for scene interpretation is a belief network in which the belief for a given scene type is computed using a set of evidence nodes which correspond to observations related to: (1) mise-en-scène, (2) identification of the subjects, (3) behavior of the subjects.

Observation of a specific behavior pattern for a set of subjects may constitute an important evidence (positive or negative) for a given scene type. This requires making specifications of the behavior attributes in the form of a prominence model. Adherence to a given prominence model is quantified by the computed prominence value. Several prominence models may be relevant to a scene. We identify the clusters which show compliance to a given prominence model as the ones which show a prominence value greater than 0.5.

We interpret a scene into two broad categories: *Subject Centric* and *Frame Centric*. Subject centric scenes contain fewer subjects with independent spatio-

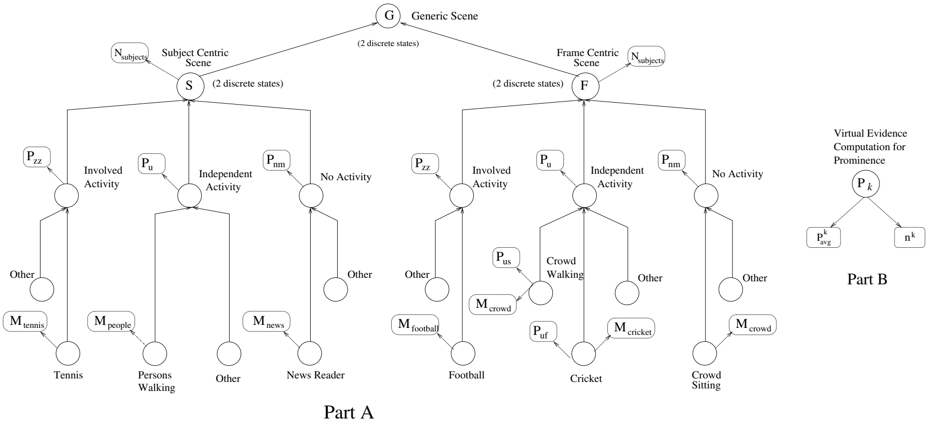
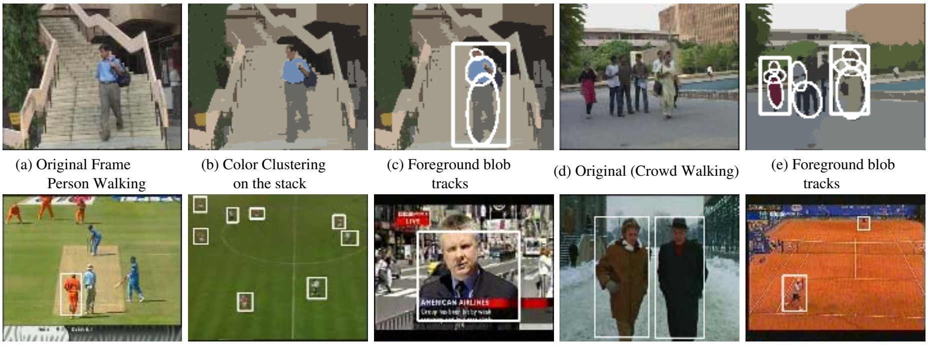


Fig. 3. Belief Network for a Scene Model



(a) Original Frame Person Walking

(b) Color Clustering on the stack

(c) Foreground blob tracks

(d) Original (Crowd Walking)

(e) Foreground blob tracks

(f) A bowling sequence (Cricket)

(g) Football scene

(h) A news reporter

(i) Persons Walking

(j) Tennis

Fig. 4. Shows some example scenes which were correctly classified by the belief network. The foreground clusters identified in each scene are shown bounded in a box.

temporal interactions. The subjects normally exist for most of the duration of the shot. Frame centric scenes contain multiple subjects that may be engaged in a group specific activity or independent activities. The overall belief in a scene is computed using a belief network shown in Fig 3 (a). The node hierarchy in the belief network is constructed in the reverse order of the scene hierarchy as in the taxonomy. In the independent-activity scenes, a subject follows an independent movement pattern without any influence from other subjects. In the involved-activity scenes, a subject normally moves in response to the movement of other subjects in the scenes. In the no-activity scenes, the subjects do not show any significant activity. Specifically we have used 4 prominence models which identify subjects with uniform motion (P_u), zigzag motion (P_{zz}), no motion (P_{nm}), and those showing a reasonable speed (P_{uf}) respectively. The scene evidence from a given prominence model is computed using a belief network shown in Fig 3(b) and comprises of the count n of the clusters which show compliance and the average prominence P_{avg} shown by the complying subjects. Different mise-en-scène cues are relevant to different scene types. These cues get instantiated as virtual evidences in the belief network. The node state which gets the highest scene probability is taken as the scene category of the given sequence.

7 Results

We have characterized different scenes by making use of key mise-en-scène aspects and subject behaviors specific to the scene types. For testing our system, we have chosen example scenes from certain sports videos. A video of any sport comprises of several types of scenes, like close-up shots of players, crowd scenes, etc. For classification purpose, we consider only those mise-en-scène cues which capture the distinguishing aspect of that game video.

We now discuss the mise-en-scène observations which we have used for different types of scenes:

Cricket: (1) Presence of grass (identified as blobs with a green hue), (2) the bowler/fielder (subject) appearing in a single color of the dress, and surrounded by grass, (3) the subjects being a humans. (4) subjects showing fast motion in a uniform direction. The last observation is common in the cricket shots in which the camera follows the ball after the batsman has played it. The camera moves fast, thus providing the subjects with an apparent fast motion in a uniform direction. Our prominence model for cricket identifies subjects which move speedily in a uniform direction. We take a cluster (subject) to be a human if the aspect ratio of its bounding box is less than 1.

Tennis: (1) the motion trajectory of a subject confined to the upper half or the lower half of the frame. (2) the subjects identified as humans (aspect ratio test) and not more than two in number, and (3) subjects adhering to the prominence model corresponding to zigzag motion. The motion of players observed in a tennis shot is primarily zigzag.

Football: The mise-en-scène observations used for football are: (1) presence of grass, (2) players distributed over the field of the view, and (3) a few of the subjects showing a zigzag motion.

News Report: We make use of the OpenCV library implementation of the face-detector to identify frontal view faces in the scene. The blobs characterizing the face region form evidence for the presence of human subject. These blobs are further grouped with other blobs below the face such that the width of the body region below the face is not more than four times the width of the face region. This kind of a grouping is likely to identify a (stationary) human body. In a news report, the reporter (subject) may stand infront of a background which may have static or moving objects (eg a traffic scene). The foreground subject in a news report adhere to a prominence model highlighting subjects with a large lifespan, and showing no motion.

Crowd Scenes: For scenes which have got frontal faces of human beings with a reasonable resolution, we use the face detector to identify face regions and then group the blobs below the face region to delineate the body region. A walking

Scene Example	Subject Centric				Frame Centric				Nodes within the relevant subcategory
	Subject Centric	Involved Activity	Independent Activity	No Activity	Frame Centric	Involved Activity	Independent Activity	No Activity	
Scene 1 (Football)	0.161	0.285	0.631	0.224	0.911	0.952	0.822	0.539	Football 0.769 Other 0.567
Scene 2 (Person walking)	0.808	0.394	0.970	0.394	0.353	0.411	0.539	0.411	PersWalk 0.852 Other 0.545
Scene 3 (Tennis)	0.884	0.955	0.815	0.383	0.391	0.534	0.537	0.410	Tennis 0.804 Other 0.553
Scene 4 (Cricket)	0.097	0.237	0.291	0.237	0.868	0.548	0.961	0.548	Cricket 0.762 Other 0.561
Scene 5 (Crowd walking)	0.164	0.225	0.284	0.628	0.877	0.546	0.960	0.607	Crowd−W 0.761 Other 0.561
Scene 6 (News Reader)	0.655	0.350	0.350	0.966	0.364	0.411	0.457	0.527	News−R 0.885 Other 0.524
Scene 7 (Crowd sitting)	0.102	0.236	0.236	0.321	0.877	0.546	0.593	0.969	Crowd−S 0.851 Other 0.544

Fig. 5. Shows the results of scene interpretation on a few example video scenes

sequence of a few subjects or several subjects(crowd) is identified by evaluating the adherence of a subject to the prominence model specifying uniform motion.

Fig 4 shows some video scene examples we have used for our experiments, and Fig 5 tabulates our results for scene interpretation. Extensibility of our interpretation framework to incorporate other classes of scenes is straightforward. The success of our framework depends crucially on the kind of observation model (evidences) used for characterizing the scenes.

8 Conclusions

As noted in the previous section, we have made use of hand-picked mise-en-scène cues and prominence models for formulating the evidences supportive for various scene types. We have achieved success in interpreting a few classes of scenes using our framework which processes and gathers evidences from raw video data. Further research is required to automatically derive the observation models for a variety of scene classes and plugging in the appropriate evidence computation modules.

References

1. Itti, L.: Models of Bottom-Up and Top-Down Visual Attention. PhD thesis, California Institute of Technology, Pasadena, California. (2000)
2. Nicolescu, M., Medioni, G.: Perceptual Grouping from Motion Cues Using Tensor Voting in 4-D . In: Proc. ECCV. Volume 3. (2002) 303 – 308
3. Shah, M., Rangarajan, K., Tsai, P.S.: Generation and Segmentation of Motion Trajectories . Volume 1. (1992) 74 – 77
4. Sarkar, S.: Tracking 2D Structures using Perceptual Organization Principles. In: Symposium on Computer Vision SCV. (95) 283 – 288
5. Sarkar, S., Majchrzak, D., Korimilli, K.: Perceptual Organization Based Computational Model for Robust Segmentation of Moving Objects . **86** (2002) 141 – 170
6. Vasseur, P., Pgard, C., Mouaddib, E.M., Delahoche, L.: Perceptual Organization Approach based on Dempster-Shafer Theory. Pattern Recognition **8** (1999) 1449 – 1462
7. Lowe, D.G.: Perceptual Organization and Visual Recognition. Boston, MA: Kluwer (1985)
8. Shashua, A., Ullman, S.: Structural Saliency: The Detection of Globally Salient Structures using locally Connected Network. (1988) 321 – 327
9. Leeuwenberg, E.L.J.: Quantification of certain visual pattern properties: Salience, Transparency, Similarity. In: Formal Theories of Visual Perception, E L J Leeuwenberg and J F J M Buffart Editors. New York: Wiley, pp. 277-298 (1978)
10. Lawton, D.T., Connell, C.C.M.: Perceputal Organization using Interestingness. In: Workshop Spatial Reasoning and Multi-Sensor Fusion. (1987) 405 – 419
11. DeMenthon, D., Megret, R.: Spatio-temporal segmentation of video by hierarchical mean shift analysis. Technical Report LAMP-TR-090, University of Maryland, College Park, MD 20742, USA (2002)
12. Wertheimer, M.: Laws of organization in perceptual forms. W. B. Ellis, editor, A Sourcebook of Gestalt Psychology. Harcourt, Brace and Company (1938)

Smooth Foreground-Background Segmentation for Video Processing*

Konrad Schindler and Hanzi Wang

Electrical and Computer Systems Engineering, Monash University
{Konrad.Schindler, Hanzi.Wang}@eng.monash.edu.au

Abstract. We propose an efficient way to account for spatial smoothness in foreground-background segmentation of video sequences. Most statistical background modeling techniques regard the pixels in an image as independent and disregard the fundamental concept of smoothness. In contrast, we model smoothness of the foreground and background with a Markov random field, in such a way that it can be globally optimized at video frame rate. As a background model, the mixture-of-Gaussian (MOG) model is adopted and enhanced with several improvements developed for other background models. Experimental results show that the MOG model is still competitive, and that segmentation with the smoothness prior outperforms other methods.

1 Introduction

A basic requirement for video processing tasks with static cameras, such as surveillance and object tracking, is to segment the objects of interest from the permanently observed background. To this end, a model is estimated which describes the background, and parts of a frame which do not fit the model within a certain tolerance are labeled as foreground. What makes the task difficult is the fact that the background dynamically changes over time. Toyama et al. have termed the task "background maintenance" to point out the dynamic aspect of keeping the model up to date, and have presented a taxonomy of possible difficulties [1]. These include gradual and sudden illumination changes, shadows, vacillating background, foreground objects which share the characteristics of the background, foreground objects which remain static and must be merged into the background model, and the situation where no training images without foreground objects are available. Examples for these difficulties can be found in the test sequences in Sect. 4.

The literature about background maintenance can be broadly classified into two main approaches. *Non-predictive* methods recover a probability density function (*pdf*) of the observations at each pixel, and classify pixels as foreground, which do not match the function. The *pdf* can be approximated by a single Gaussian [2], a mixture of Gaussians [3] or a non-parametric distribution [4]. Some authors use not only intensities, but also higher-level information such as optical

* KS has been funded by the *Monash Institute for Vision Systems*, HW by the *ARC*.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 581–590, 2006.

flow [5]. A few methods do not work on single pixels: in [6], the background model is compressed to a set of codebook vectors, while [7] uses a simple mean image as background model, and normalized cross-correlation of small windows to measure how well two regions match.

A second class of methods uses *prediction* rather than density estimation to predict the pixel value, and classifies pixels as foreground, which do not match the prediction. Linear prediction is the basis of [1]. That paper also introduced the notion that background maintenance has to take into account different spatial scales: the initial result is improved using information at region-level for hole-filling, and at frame-level by maintaining several background models and switching between them, such that the foreground does not become too large. Prediction can also be performed with a Kalman filter [8], through projection onto a PCA-basis [9], or with an autoregressive model [10].

A classical statistical model, which is able to deal with many difficulties, is the mixture-of-Gaussian (MOG) model introduced by Stauffer and Grimson [3]. It describes the values of each background pixel throughout the sequence with a mixture of Gaussian distributions. Since several Gaussians are used, it correctly models multi-modal distributions due to periodic changes (e.g., a flag in the wind or a flickering light source), and since the parameters of the Gaussians are continually updated, it is able to adjust to changing illumination, and to gradually learn the model, if the background is not entirely visible in the beginning.

A straight-forward implementation of the MOG method has been shown to fail on several of the difficulties described above [1]. One goal of this paper is to show that most of these failures can be avoided, if the improvements suggested for different other background maintenance algorithms are incorporated into the MOG model, too. If the method is implemented carefully, the results are at least as good as for other standard methods. Firstly, the difficulties due to shadows and highlights can be solved using chromaticity coordinates, as already proposed in [11]. The second difficulty is more deep-rooted: the method uses a single learning rate to control two distinct phenomena, the adaptation to *changing illumination* and the fading of *static foreground objects* into the background. Therefore, foreground objects which stop moving are absorbed into the background too quickly. To overcome this limitation, a learning delay is introduced, which explicitly states how long a static object should remain foreground. Thirdly, we show that information, which can only be detected at frame-level (e.g. sudden changes in global illumination), can easily be fed back into the MOG model via the learning rate.

The main contribution of the paper does not concern the maintenance of the background model itself, but the way it is used to label pixels as background or foreground. Commonly, the background likelihood is simply thresholded for each pixel independently. In contrast, we argue that even at a low level the field of background probabilities contains spatial information. For a long time researchers have recognized that even prior to any semantic interpretation the visual world is smooth, in the sense that an image is generated by objects which are mapped to image regions with common properties [12]. This does not require

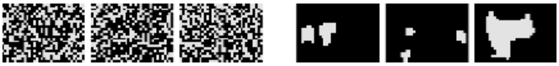


Fig. 1. Smoothness as prior belief. Random samples from the posterior distributions of segmentations without (left) and with (right) smoothness prior. Background probabilities are uniformly distributed, there is no semantics. Still the patterns on the right are visually more realistic.

semantic interpretation – even if the objects are unknown, the world is *a priori* more likely to generate a smooth foreground/background pattern than a random pattern (see Fig. 1). To make full use of the estimated likelihoods and add a smoothness prior, we cast the foreground/background segmentation as a labeling problem on a first-order Markov random field (MRF), and show how its optimal configuration can be efficiently found.

The approaches closest to ours probably are [13], and very recently [14]. The former also model smoothness with a MRF. In their posterior, they combine normalized color and intensity (as advocated in Sect. 2), conventional (R, G, B)-color, and the output of an edge detector. For the resulting complicated energy functional, only a minimum of undetermined goodness is found. We propose a simpler posterior, which uses less information, but can be globally optimized and requires fewer parameters. [14] model both position and appearance in a single *pdf*, estimated with a kernel density method. They also estimate a foreground distribution, assuming smoothly changing foreground, and use a MRF-formulation similar to the one presented here to enforce spatial coherence.

In the last section, experiments on the *Wallflower* benchmark are presented, which show that the enhanced MOG-model is competitive with all other background maintenance methods we are aware of, and that, when used with a smoothness prior, it outperforms all other tested methods.

2 The Mixture-of-Gaussian Model

Principle. The intuition behind the MOG-model is the following: the intensities $\vec{x}$ of a given pixel form a time series, which can be represented as the mixture of a small number of Gaussians. Let the maximum number of Gaussians for a pixel be K (in our implementation set to $K = 5$). The probability that a pixel assumes a value $\vec{x}$ at a certain time t is then given by [3]

$$P(\vec{x}_t) = \sum_{i=1}^{K} \frac{w_{i,t}}{\sqrt{(2\pi)^n |\mathtt{S}_{i,t}|}} e^{-\frac{1}{2}(\vec{x}_t - \vec{m}_{i,t})^\mathsf{T} \mathtt{S}_{i,t}^{-1} (\vec{x}_t - \vec{m}_{i,t})} \tag{1}$$

where $\vec{m}_i$ is the mean of the i^{th} Gaussian, $\mathtt{S}_i$ is its covariance matrix, and w_i is its weight (the portion of data it accounts for), all at time t. For computational reasons, the channels of the image are assumed to be independent, so that $\mathtt{S}_k = \text{diag}(\vec{s}_k^{\,2})$. To determine how many of the K Gaussians are needed for a pixel, the Gaussians are sorted by $\frac{w_k}{\text{mean}(\vec{s}_k)}$, meaning that distributions based on a lot of evidence and distributions with low uncertainty come first. Only the first B distributions are chosen to represent the background, where

$$B = \arg\min_{b} \left(\sum_{k=1}^{b} w_k > T \right) \tag{2}$$

The value T determines the minimum fraction of the recent data at the location $\vec{x}$, which should contribute to the background model. If the background distribution is complicated, a larger value is needed to ensure enough Gaussians to approximate it. We use $T = 0.9$.

The parameters of the model are estimated in an initial training phase, and then continually updated as new data is observed. If the new pixel value $\vec{x}_t$ belongs to the i^{th} distribution, the parameters are updated to

$$\begin{aligned} \vec{m}_{i,t} &= (1-\alpha)\vec{m}_{i,t-1} + \alpha\vec{x}_t \\ \vec{s}^{\,2}_{i,t} &= (1-\alpha)\vec{s}^{\,2}_{i,t-1} + \alpha(\vec{x}_t - \vec{m}_{i,t})^{\mathsf{T}}(\vec{x}_t - \vec{m}_{i,t}) \end{aligned} \tag{3}$$

Here, α is the learning rate, which determines, how fast the parameters are allowed to change. The weights are updated to

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha U_{k,t} \quad , \quad U_{k,t} = \begin{cases} 1\ldots \text{if } i = k \\ 0\ldots \text{else} \end{cases} \tag{4}$$

Since new data gradually replaces older data in the background model, the algorithm can deal with gradual changes of the background, such as the ones typically encountered with natural light.

Implementation Issues. After its appearance in the literature, the MOG-model has been criticized by proponents of other background models, based on failure in a number of experiments. In this section we will argue that the MOG-model performs at least as well as other state-of-the-art methods, if it is carefully implemented. A quantitative comparison is presented in Sect. 4.

A frequent problem of background modeling methods is that cast shadows and moving highlights are incorrectly labeled as foreground, because they induce a sudden change of brightness. The common assumption to deal with these situations is that a change in illumination intensity alters only the lightness, but not the color of the region [15]. To suppress the influence of the lightness, several background modeling methods use normalized chromaticity coordinates, e.g. [4, 5]. The normalized chromaticity values are defined by $(r, g, b) = \frac{1}{R+G+B}(R, G, B)$, where two of the three values are sufficient. As a third coordinate, the intensity $I = (R + G + B)$ is used, which otherwise would be lost. In the new colorspace (r, g, I), color and intensity have been separated, and a shadow or highlight is expected to alter only the intensity. In any environment with a diffuse lighting component or multiple light sources, a shadow will only occlude a certain portion of the light, and a similar argument can be made for a highlight. Hence, the change in intensity is expected to stay within a certain range, $\beta \leq I_t/I_{t-1} \leq \gamma$. Within that range, the distribution is *not* Gaussian.

Translated to the MOG-model, where we have to deal with multiple modes, and the expectation of the previous intensity is the mean $m_{\mathbf{I}i}$, we get the condition $\beta \leq I_t/m_{\mathbf{I}i} \leq \gamma$. Empirically, the intensity change due to shadows and highlights is at most 50%, so $\beta = 0.6, \gamma = 1.5$. In [11], a more exact procedure is derived based on statistical hypothesis-testing. However, we found that our simple approach gives good results and thus avoid the hypothesis test, which may be particularly vulnerable to the simplifying assumption of independence between the color channels.

Another issue when using the MOG-model is that the gray-value distribution is at best approximately Gaussian, so that the standard deviations $\vec{s}$ may be estimated incorrectly. On one hand, the sensor accuracy is limited, so extremely small standard deviations do not make sense. On the other hand, each Gaussian accounts only for one mode of the distribution, so $\vec{s}$ should only account for the variation within that mode. It is a matter of good engineering to bound $\vec{s}$ to reasonable values. In our implementation, we use $2 < s_{\mathbf{r},\mathbf{g}} < 15$ (for 8-bit images).

Thirdly, there is a dilemma how to set the correct learning rate. If a low α is chosen, the background model will take too long to adapt to illumination changes, while a high α will quickly merge the objects of interest into the background when they stop or move slowly. The reason is that a single learning rate is used to cover two different phenomena, namely the smooth variation of the background process over time, and the transition from foreground to background. This transition is a discrete process depending on the user's requirements ("after how many frames shall a static foreground object become background?"). A straight-forward way to separate the two phenomena is to stop learning a pixel's process, when it becomes foreground. After the pixel has continuously remained in the foreground for a given number of frames, background learning with equations (3) and (4) continues, and it will fade into the background with the speed given by the learning rate, if it remains static.

Finally, Toyama et al. have used a long-term memory to maintain multiple background models and switch between them to cope with sudden changes, such as switching on the light in a room. We agree with their reasoning that information at the frame level, rather than pixel level, is required to detect this type of change. The MOG-model provides an elegant way to deal with such situations: if a global change occurs, and almost the entire image is labeled as foreground, increasing the learning rate will automatically boost the adaptation to the new global conditions.

3 Adding Smoothness

In most probabilistic background models, each pixel is considered independent of the others, and a binary decision is taken: if the pixel does not match any of the background distributions, it is labeled as foreground. This contradicts the well-known fact that the world consists of spatially consistent entities, often called the *smoothness* assumption. In fact, standard background modeling algorithms such as the original MOG-method or *Wallflower* use an ad-hoc version of the smoothness assumption: they clean the foreground/background segmentation by deleting small foreground clusters using connected components.

We propose a more principled way to incorporate a smoothness prior: rather than simple thresholding, a continuous background probability value is retained for each pixel, and the foreground segmentation is treated as a labeling problem on a first-order Markov random field. Maximizing the posterior probability then results in a smooth, and more correct, segmentation.

Markov Random Fields (MRF) are a probabilistic way of expressing spatially varying priors, in particular smoothness. They were introduced into computer vision by Geman and Geman [16]. A MRF consists of a set of sites $\{x_1 \dots x_n\}$ and a neighborhood system $\{N_1 \dots N_n\}$, so that N_i is the set of sites, which are neighbors of site x_i. Each site contains a random variable U_i, which can take different values u_i from a set of labels $\{l_1 \dots l_k\}$. Any labeling $U = \{U_1 = u_1 \dots U_n = u_n\}$ is a realization of the field. The field is a MRF, if and only if each random variable U_i depends only on the site x_i and its neighbors $x_j \in N_i$. Each combination of neighbors in a neighborhood system is called a *clique* C_{ij}, and the prior probability of a certain realization of a clique is $e^{-V_{ij}}$, where V_{ij} is called the *clique potential.* The basis of practical MRF modeling is the Hammersley-Clifford Theorem, which states that the probability of a realization of the field is related to the sum over all clique potentials via $P(U) \propto \exp(-\sum V_{ij}(U))$.

If only cliques of 1 or 2 sites are used, the field is called a first-order MRF. The 1-site clique for each x_i is just the site itself, with likelihood $e^{-W_i(u_i)}$. Each 2-pixel clique consists of x_i and one of its neighbors, and has the likelihood $e^{-V_{ij}(u_i,u_j)}$. Following Bayes' theorem, the most likely configuration of the field is the one which minimizes the posterior energy function

$$E(U) = \sum_{x_i} \sum_{x_j \in N_i} V_{ij}(u_i, u_j) + \sum_{x_i} W_i(u_i) \tag{5}$$

It remains to define the clique potentials V_{ij}. If the goal is smoothness, and the set of labels does not have an inherent ordering, a natural and simple definition is the *Potts model* [17]

$$V_{ij} = \begin{cases} d_{ij} & \text{if } u_i \neq u_j \\ 0 & \text{else} \end{cases} \tag{6}$$

If two neighboring sites have the same label, the incurred cost is 0, else the cost is some value d_{ij}, independent of what the labels u_i and u_j are.

Application to Background. In the following, we will convert the background modeling problem into an MRF and show how to efficiently solve it. First, we have to define a background likelihood for each pixel. In the conventional MOG-method, a pixel $\vec{x} = [x_\mathbf{r}, x_\mathbf{g}, x_\mathbf{I}]^\mathsf{T}$ in the current frame is labeled as foreground, if it is far away from all modes of the background in terms of color or intensity.

$$\vec{x} \rightarrow \mathcal{F} \text{ if } \begin{cases} \frac{(x_{\mathbf{r}i} - m_{\mathbf{r}i})^2}{s_{\mathbf{r}i}^2} + \frac{(x_{\mathbf{g}i} - m_{\mathbf{g}i})^2}{s_{\mathbf{g}i}^2} > \theta^2 & \forall i \in \{1..K\} \quad \textbf{or} \\ \frac{x_\mathbf{I}}{m_{\mathbf{I}i}} < \beta \textbf{ or } \frac{x_\mathbf{I}}{m_{\mathbf{I}i}} > \gamma & \forall i \in \{1..K\} \end{cases} \tag{7}$$

In other words: $\vec{x}$ matches the i^{th} Gaussian, if its normalized distance from the mean is below a threshold θ (to cover 99.5% of the inliers to a Gaussian, $\theta = 2.81$). The evidence that $\vec{x}$ belongs to the background $\mathcal{B}$ is the probability that it belongs to the Gaussian, which it fits best, and only those Gaussians are valid, for which the intensity difference is not too large.

It is easy to convert this condition into a likelihood. The cost for labeling a pixel as foreground is constant, and shall be lower than the cost for labeling it as background only if condition (7) does not hold. The negative log-likelihood (the cost) of $\vec{x}$ in the i^{th} Gaussian is

$$W_i(\vec{x}) = \begin{cases} \frac{(x_{\mathbf{r}} - m_{\mathbf{r}i})^2}{s^2_{\mathbf{r}i}} + \frac{(x_{\mathbf{g}} - m_{\mathbf{g}i})^2}{s^2_{\mathbf{g}i}} & \text{if } \beta \leq \frac{x_{\mathbf{I}}}{m_{\mathbf{I}i}} \leq \gamma \\ a\theta^2 & \text{else} \end{cases} \tag{8}$$

where a is a constant >1, stating that the background cost is higher than the foreground cost, if the intensity difference is large Empirically, $a = 2.5$ performs satisfactory for all image sequences we have tested. Among the K Gaussians, the strongest evidence that $\vec{x}$ belongs to the background is the one with the lowest cost. If the modes are well separated, the likelihood of belonging to any other Gaussian is small, so the cost of assigning $\vec{x}$ to the background/foreground is

$$\begin{aligned} W(\vec{x} \in \mathcal{B}) &= \arg\min_i (W_i(\vec{x})) \\ W(\vec{x} \in \mathcal{F}) &= \theta^2 \end{aligned} \tag{9}$$

To model the neighborhood, we use the simplest possible definition: a pixel is connected to each neighbor in its 4-neighborhood, and the clique potential is a constant, which determines the amount of smoothing. We write the constant $V_{ij} = b\theta^2$, so that the cost for large intensity differences in equation (8) and the clique potential are on the same scale. Useful values are $1 \leq b \leq 4$.

Maximizing the posterior likelihood of the MRF is equivalent to minimizing the energy functional (5) over the space of realizations of the MRF. Since our special case has only 2 labels (background and foreground), the *global minimum* can be found in low polynomial time with the min-cut algorithm [18]: the MRF is converted into a graph, where the sites x_i are the nodes, and the cliques C_{ij} are the arcs joining the nodes x_i and x_j, with cost V_{ij}. The graph is augmented with two extra nodes for the two labels, which are connected to every site by an arc representing the corresponding likelihood W_i (plus a constant larger than the maximum clique potential for one node). The minimum cut on this graph partitions it into two sub-graphs, such that each node is only connected to one label. Min-cut is very efficient: we have tested it with the *Wallflower* benchmark with image size 160×120 pixels (see Sect. 4 for results). On a 2 GHz desktop PC, constructing the graph, solving the optimization, and clearing the memory takes on average 14 milliseconds, and thus does not impair the real-time capabilities of the MOG method.

4 Experimental Results

The algorithm has been tested with the *Wallflower* benchmark. This data set has been used by Toyama et al. to assess a large number of background maintenance methods. It has also been used by Kottow et al. to assess their method [6]. The data set consists of 7 video sequences of resolution 160×120 pixels, each representing a different type of difficulty that a background modeling system may meet in practice. For the last used frame of each sequence, manually segmented ground truth is available to enable a quantitative comparison. Tab. 1 shows the number of foreground pixels labeled as background (false negatives - FN), the number of background pixels labeled as foreground (false positives - FP), and the total percentage of wrongly labeled pixels $\frac{FN+FP}{160\times120}$. Furthermore, the total number and percentage of wrongly labeled pixels over all 7 difficulties is given. As explained above, the authors of *Wallflower* have noted that information at the frame level is needed to deal with sudden illumination changes. However, they do not seem to have included this information in their implementations of other tested algorithms. This distorts the comparison, hence we also display the total results without the **Light Switch** sequence (column TOTAL*).

We have presented two improvements. First, we have shown that the original MOG-method is a valid and competitive algorithm, if implemented with the same care as other methods, and secondly we have applied the MRF-concept as a sound way to incorporate spatial smoothness. To separate the two parts' contributions, we present the results of our MOG algorithm cleaned up with the conventional connected component method, and the improved results using MRF smoothing. We did not tune towards the single sequences. All parameters were kept constant, except for the (automatic) increase of the learning rate in case of a sudden illumination change, as explained above. For some practical applications it may be possible to exclude certain scenarios and empirically find

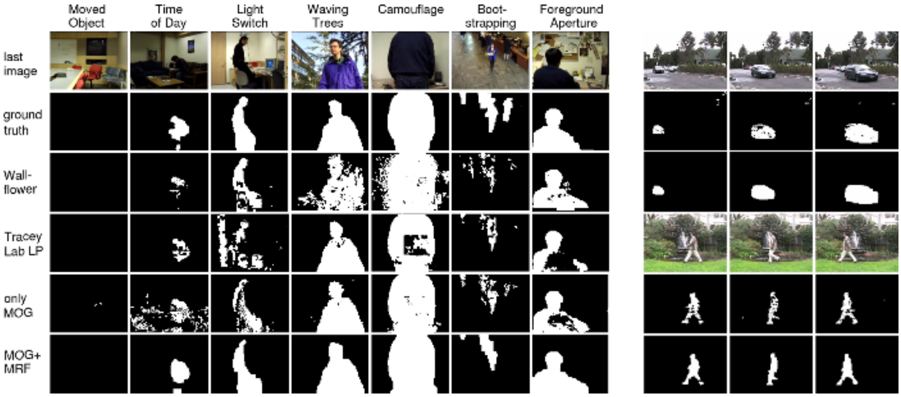


Fig. 2. ***Left:*** Wallflower benchmark. ***Right:*** "Car" and "fountain" videos. 3 frames of each sequence, results of improved MOG, results of MOG with MRF smoothing.

Table 1. Wallflower benchmark. † were reported in [1], ‡ were reported in [6].

Algorithm	errors	MO	TOD	LS	WT	C	B	FA	TOTAL	TOTAL*
Eigen-	FN	0	879	962	1027	350	304	2441		
background†	FP	1065	16	362	2057	1548	6129	537	17677	16353
	%	*5.6*	*4.7*	*6.9*	*16.1*	*9.9*	*33.5*	*15.5*	*13.2*	*14.2*
MOG	FN	0	1008	1633	1323	398	1874	2442		
(original)†	FP	0	20	14169	341	3098	217	530	27053	11251
	%	*0.0*	*5.4*	*82.3*	*8.7*	*18.2*	*10.9*	*15.5*	*20.1*	*9.8*
Wallflower†	FN	0	961	947	877	229	2025	320		
	FP	0	25	375	1999	2706	365	649	11478	10156
	%	*0.0*	*5.1*	*6.9*	*15.0*	*15.3*	*12.5*	*5.1*	*8.5*	*8.8*
Tracey	FN	0	772	1965	191	1998	1974	2403	12035	8046
Lab LP‡	FP	1	54	2024	136	69	92	356		
	%	*0.0*	*4.3*	*20.8*	*1.7*	*10.8*	*10.8*	*14.4*	*9.0*	*7.0*
this paper	FN	0	203	1148	43	110	1159	1023	7340	5628
(only MOG)	FP	19	1648	564	278	468	143	534		
	%	*0.1*	*9.6*	*8.9*	*1.7*	*3.0*	*6.8*	*8.1*	*5.5*	*4.9*
this paper	FN	0	47	204	15	16	1060	34	3808	3058
(MRF smoothed)	FP	0	402	546	311	467	102	604		
	%	*0.0*	*2.3*	*3.9*	*1.7*	*2.5*	*6.1*	*3.3*	*2.8*	*2.7*

better parameter settings. However, we have found that this is not critical. The overall performance only increases by ≈600 pixels (15%), even if the optimal values are chosen for each sequence separately (which of course is improper tuning towards a specific data set).

Figure 2 shows the segmentation results for the most successful algorithms on the *Wallflower* data. A quantitative comparison is given in Tab. 1. The comparison should be taken with a grain of salt: choosing an algorithm will depend on the expected difficulties in a given application. Note however that our method yields the best result for all sequences. Also, an actual implementation must take into account the nature of the application. For example, in a high-security setting, one will seek to minimize false negatives and rather accept more false alarms. Any of the given algorithms has a parameter, which governs its sensitivity (up to which distance from the model a pixel is assigned to the background), and can be tuned accordingly. Two more results of our method are shown in Fig. 2: a car moving in front of waving trees, and a person walking past a fountain, which is similar in color to the person's clothing.

5 Conclusions

A framework for smooth foreground/background segmentation in video streams has been presented, which can be applied with any probabilistic background model. In the present work, an improved MOG method is used, which overcomes a number of problems of the original method. The assumption of a smooth foreground/background pattern is treated in a principled, but computationally tractable way: segmentation is cast as a labeling problem on a particularly simple Markov random field, and solved with a classical algorithm.

It has been demonstrated that the method is fast enough for video-processing, and that it outperforms methods, which neglect smoothness or incorporate it in an ad-hoc manner. We do not challenge the principle formulated by Toyama et al., that semantic segmentation should not be handled by a low-level module like background maintenance. Rather, we claim that spatial smoothness already is a guiding principle before semantic interpretation.

References

1. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In: Proc. 7th ICCV. (1999) 255–261
2. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: Real-time tracking of the human body. IEEE TPAMI **19** (1997) 780–785
3. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proc. IEEE CVPR. (1999) 246–252
4. Elgammal, A., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: Proc. 6th ECCV. (2000) 751–767
5. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: Proc. IEEE CVPR. (2004) 302–309
6. Kottow, D., Koppen, M., Ruiz del Solar, J.: A background maintenance model in the spatial-range domain. In: Proc. 2nd SMVP. (2004)
7. Matsuyama, T., Ohya, T., Habe, H.: Background subtraction for non-stationary scenes. In: Proc. 4th ACCV. (2000) 662–667
8. Koller, D., Weber, J., Malik, J.: Robust multiple car tracking with occlusion reasoning. In: Proc. 3rd ECCV. (1994) 189–196
9. Oliver, N., Rosario, B., Pentland, A.: A Bayesian computer vision system for modeling human interactions. In: Proc. ICVS. (1999)
10. Monnet, A., Mittal, A., Paragios, N., Ramesh, V.: Background modeling and subtraction of dynamic scenes. In: Proc. 9th ICCV. (2003) 1305–1312
11. Greiffenhagen, M., Ramesh, V., Niemann, H.: The systematic design and analysis cycle of a vision system: a case study in video surveillance. In: Proc. IEEE CVPR. Volume 2. (2001) 704–711
12. Poggio, T., Torre, V., Koch, C.: Computational vision and regularization theory. Nature **317** (1985) 314–319
13. Paragios, N., Ramesh, V.: A MRF-based approach for real time subway monitoring. In: Proc. IEEE CVPR. (2001)
14. Sheikh, Y., Shah, M.: Bayesian object detection in dynamic scenes. In: Proc. IEEE CVPR. (2005) 74–81
15. Levine, M.D.: Vision in Man and Machine. McGraw-Hill (1985)
16. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. IEEE TPAMI **6** (1984) 721–741
17. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: Proc. IEEE CVPR. (1998) 648–655
18. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In: Proc. 3rd EMMCVPR. (2001)

Efficient Object Segmentation Using Digital Matting for MPEG Video Sequences

Yao-Tsung Jason Tsai and Jenn-Jier James Lien

Robotics Lab., Dept. of Computer Science and Information Engineering,
National Cheng Kung University, No.1, Ta-Hsueh Road, Tainan 701, Taiwan
{kenko, jjlien}@csie.ncku.edu.tw
http://robotics.csie.ncku.edu.tw

Abstract. We developed an automatic object segmentation system to separate the foreground objects from the background scene in the MPEG video sequence. The system consists of two modules: the background modeling and updating module and the foreground object extraction module. For the first module and comparing to existing methods, the background model can be constructed no matter whether there exist moving foreground objects or not. In addition, the background model is capable of handling the illumination changes and intrusive but motionless targets by using the short-term approach and long-term approach, respectively, to keep updating the background model. For the second module, the noises and shadows are eliminated and the holes are filled in order to reduce the false and the missing foreground detection components, respectively. Furthermore, one particular function in this module is the automatic digital matting, which can be applied to have visually accurate segmentation result for the foreground objects.

1 Introduction

Video segmentation, which separates the foreground objects from the background scene in the video sequence, is one of the essential works for video compression like MPEG. Because the background regions are more stable than the foreground objects over the video sequence, so to have the efficient and fast compression process, it is necessary to process the foreground objects and the background regions individually.

Background subtraction is a typical method to extract the moving foreground objects in image sequences. For recent works of background subtraction, a Gaussian model [7] or mixture of Gaussian models (MOGS) [6], [10] are used. But they may not get the complete foreground regions and shadow effects may still exist. Especially, the approaches of these papers need a period of time without any moving objects in the beginning of video sequence to model the background. The work in [3] proposed a predictive watershed method to improve the segmentation result of using background subtraction method. But the boundaries of foreground objects do not segment precisely.

To solve above existing problems, we developed an automatic object segmentation system, which consists of two modules, as shown in Fig. 1. The first module is the background modeling and updating module, which can construct the background model with or without existing moving objects. The background model will also be

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 591 – 601, 2006.

updated to adapt to the illumination variances of the background scene and the intrusive but motionless targets of the foreground objects by the short-term approach and the long-term approach, respectively. The second module is the foreground object extraction module, which not only eliminates the noises and the shadow regions, but also applies automatic digital matting approach to optimize the segmentation results of the objects' boundaries. In addition, in this work, all image sequences were shot by one single video camera, which was supported by a tripod without any pan-tilt-zoom movements and has the frame rate as 30 frames per second.

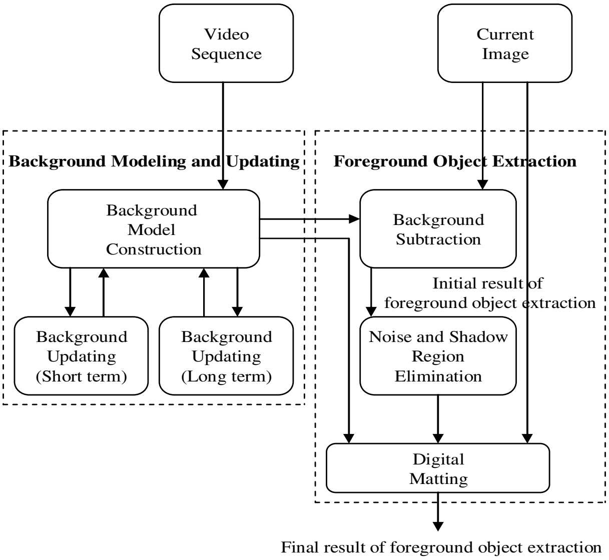

Video Sequence
Current Image
Background Modeling and Updating
Foreground Object Extraction
Background Model Construction
Background Subtraction
Initial result of foreground object extraction
Background Updating (Short term)
Background Updating (Long term)
Noise and Shadow Region Elimination
Digital Matting
Final result of foreground object extraction

Fig. 1. The flowchart of the object segmentation system in a video sequence

2 Background Modeling and Updating

The background modeling and updating module of the proposed system is modified the works in [7], [6], and [10]. This module is built based on the Gaussian model and is capable of handling the illumination changes and intrusive but motionless targets by using the short-term approach and long-term approach.

2.1 Background Model Construction

We will construct a background model no matter whether there exist foreground objects or not in the beginning of the video sequence. Initially, we make an assumption that the appearance frequency of a pixel belonging to the background region must be higher than that of this pixel belonging to the foreground objects in the first M consecutive frames. Then let V^M be a matrix containing M consecutive images, each of which is a $W*H$-pixel (width * height = 320*240) image vector; and V^i (x, y) is the intensity value of a pixel located at (x, y) in the ith image of V^M. Thus, the intensity distribution of a pixel $V(x, y)$ over the first M (M=60) consecutive images (as shown in Fig. 2.a) at one of the RGB channels can be represented as shown in Fig. 2. b. Based on the median value $\lambda(x, y)$ and the standard deviation $\sigma(x, y)$ of the intensity distribution for the pixel $V(x, y)$, $V^M(x, y)$ is classified as stationary pixels, where

$$| V^{M} (x, y) - \lambda(x, y) | < 2 * \sigma(x, y) \tag{1}$$

Subsequently, the average value $\mu(x ,y)$ of $V^M(x, y)$ is calculated and assigned to the background model for the pixel located at location (x, y). Based on the same process for the entire W*H-pixel image over the M consecutive images, the background model can be constructed for each video sequence, as shown in Fig. 2.c.

The reason why the median value is chosen instead of the mean value for the initial process is because median value is not sensitive to noises, such as the appearances of the moving foreground objects or sudden illumination change, as shown in Fig. 2.b. In addition, our assumption is adjustable. That is, if the background scene is stable, then the appearance frequency of the pixel belonging to the background region can be higher. On the contrary, if the background scene is unstable, such as affected by slowly moving foreground objects or unstable lightings, then the appearance frequency of the pixel needs to be lower.

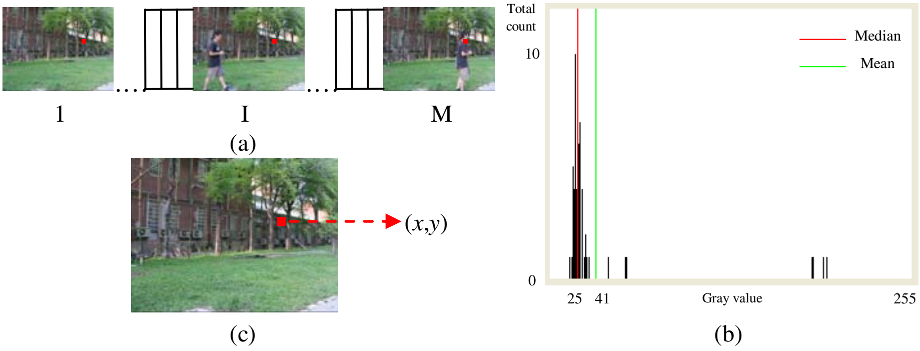


Fig. 2. Background model construction. (a) The first M(M=60) consecutive images in the video sequence. (b) The intensity distribution of M consecutive images for a pixel located at (x,y). (c) The background image (or model) is built based on the average value μ (x,y) of v^z (x,y) located at location (x,y).

2.2 Background Updating

The initial background model cannot be expected to work for long periods of time. Without updating the background model, it will cause the false extraction (or detection) results by illumination changes, such that the sun is blocked by clouds and then appears again, or physical changes, such that an object is deposited or a car is parked. To overcome above problems, two different approaches are applied to update the background model: the short-term updating approach and the long-term updating approach.

For the short-term updating approach, this approach will quickly update the background pixels in order to maintain the sensibility of extracting or detecting the moving foreground pixels. The intensity of the background pixel $\mu_t(x, y)$ at location (x, y) and time t is updated by following equation:

$$\mu_t(x,y) = \begin{cases} (1-\rho) * \mu_{t-1}(x,y) + \rho * \mu_t(x,y) & \text{if } \mu_t(x,y) \in \text{Background} \\ \mu_{t-1}(x,y) & \text{if } \mu_t(x,y) \in \text{Foreground} \end{cases} \tag{2}$$

where ρ is the updating rate. If the ρ value is higher, then the background pixel is easier to adapt to the new environment. For example, it is necessary to have the higher ρ value when the branches or leaves of trees move periodically by wind in the background scene. Furthermore, according to our experience, if more than 80% pixels of the image are suddenly classified as foreground pixels for longer than 5~15 consecutive frames, then this factor is not caused by flashlight and the background model will start to reconstruct completely from the beginning.

For the long-term updating approach, this approach will update to the background pixels when the moving foreground object is detected for a long time without any motion. Then the region of this foreground object should be updated into background pixels. We apply the stationary map method in [2] and [7] to this approach. That is, each pixel has its corresponding value in the stationary map (Sp). The corresponding value of the foreground pixel in the stationary map will increase by one if its intensity difference between two consecutive input images is less than the threshold value (*th*). Otherwise, the corresponding value will be reset to zero if this pixel is not stationary. Following equation represents this approach:

$$Sp_t(x,y) = \begin{cases} Sp_{t-1}(x,y) + 1 & \text{if } \quad |I_t(x,y) - I_{t-1}(x,y)| < th; \\ & \quad \text{and } I_{t\text{-}1} \text{ and } I_t \in \text{Foreground} \\ 0 & \quad \text{otherwise} \end{cases} \tag{3}$$

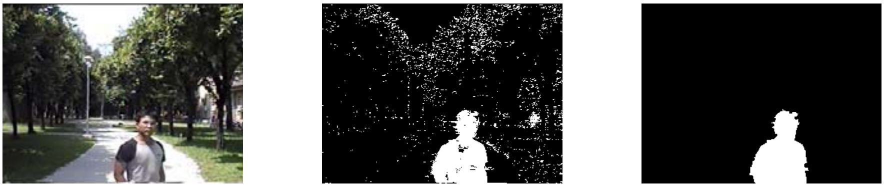
Therefore, if the corresponding value of the foreground pixel in the stationary map is bigger than a predefined value, then this pixel will be updated to the background model with the current intensity value.

3 Foreground Object Extraction

After the background model is constructed, the foreground object extraction module can be applied to visually accurately extract the foreground objects. This module consists of three approaches: the background subtraction, noise and shadow region elimination, and digital matting.

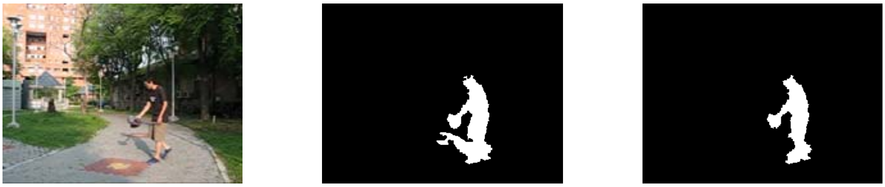
3.1 Background Subtraction

Based on the background model, the background subtraction approach is applied to initially separate the foreground objects from the background scene. Each pixel of the input image in the video sequence is compared with the corresponding pixel at the background model. If the absolute difference of the intensity values between both pixels is larger than a predefined value, then this pixel is extracted to be the candidate foreground pixel. Otherwise, this pixel will be assigned to follow the process in the background modeling and updating module. Some example results of the background subtraction process are shown in Fig. 3.a and 3.b and Fig. 4.a and 4.b.



(a) (b) (c)

Fig. 3. Noise region elimination. (a) The original image. (b) The result of the background subtraction. There still exist a lot of noises, which are caused by illumination variances or small movements of tree branches or leaves. (c) The result after eliminating the noise region.



(a) (b) (c)

Fig. 4. Shadow region elimination. (a) The original image. (b) The result of the background subtraction with shadow effects. (c) The result after eliminating the shadow region.

3.2 Noise and Shadow Region Elimination

As shown in Fig. 3.b and Fig. 4.b, simply applying the background subtraction process cannot perform accurate result of separating the foreground pixels from the background pixels. The existing problems include the false detection foreground pixels and missing detection foreground pixels. The situations of the false detection foreground pixels include the white noises, especially, in the outdoor environment (see Fig. 3.b), the reflection of light, small movements of tree branches or leaves in the background scene (see Fig. 3.b), and the shadows (see Fig. 4.b). The situation of the missing detection foreground pixels is the foreground region with holes, which are caused by the similar intensity values between foreground pixels and background pixels (see Fig. 3.b).

This work applies the morphological operations including the erosion, dilation and connected-component labeling (CCLabeling) [9] operations to eliminate those false detection foreground pixels excepting those pixels belonging to the shadows and to fill the holes inside the foreground region, as shown in Fig. 3.c. The shadow region is easily detected as the foreground region and it always follows the movement of the foreground object, so it is a challenging work to discriminate between the foreground object region and the background shadow region. Here, this work can successfully reduce the shadow effects by using the method of the vector model proposed in [5]. That is, each detected foreground pixel P is the center pixel of one 3×3-pixel window, and there are 8 pixels in its 8-connected neighboring area. Those 9 pixels located in the two-dimensional (2D) window are then regarded as one intensity column vector. Because shadow is usually caused by illumination factor, so if the pixel locates in the shadow region, then the intensity vectors of this pixel exist a linear dependent relation between the shadow region and the corresponding background region. Conversely, the linear dependent relation does not exist for any foreground pixels. By using this method, the shadow region is successfully removed, as shown in Fig. 4.c.

3.3 Digital Matting

Because of the background subtraction process and morphological operations, the segmentation results of the foreground objects are fragmentary, especially, in the boundaries. In order to improve the segmentation performance, the digital matting approach is applied in this system. In digital matting, a foreground component is extracted from a background scene by estimating a color and opacity for the foreground component at each pixel. That is, each pixel of the input image *C* is a composite of the foreground color *F* and the background *B* by the compositing equation:

$$C = \alpha F + (1 - \alpha) B \tag{4}$$

where α is the opacity value between zero and one. Using this opacity value α, we can form the composite *C* as a linear combination of *F* and *B*. By calculating the best parameters of α, *F*, and *B* of the composite *C*, we can extract the foreground components accurately from the input image. Modern approaches [1], [4] that work with natural images often require user to manually segment each image into three regions "background," "foreground," and "unknown." For a long video sequence, it will cause a lot of loading to users. In order to build an automatic object segmentation system in video sequences, the proposal system needs to be capable of automatically segmenting the image into foreground, background, and unknown regions as the segmented image of digital matting, called matting mask (see Fig. 5.c).

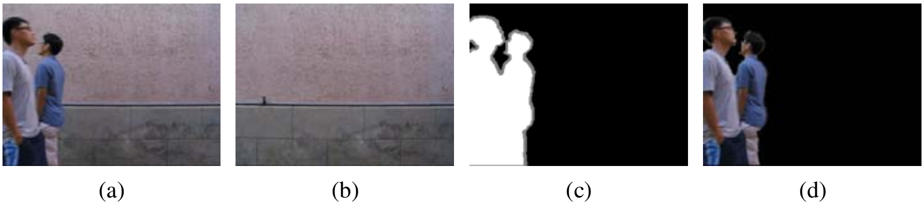


Fig. 5. Foreground object segmentation using the digital matting approach. (a) The original image. (b) The background model. (c) The initial result of extracting foreground objects, which contain foreground (white color), background (black color), and unknown (gray color) regions. (d) The result after digital matting.

The flowchart of the automatic digital matting process is shown in Fig. 6. Digital matting needs to use the image, which has been segmented into three regions: "background," "foreground," and "unknown." So we directly use the result of the background subtraction (including the morphological operations) to provide a segmented image for digital matting process instead of segmenting image by user (see Fig. 5.a and c). Based on the extracted foreground objects, we apply erosion operation to shrink the object regions. All pixels inside the shrunken object regions are defined as the "foreground"

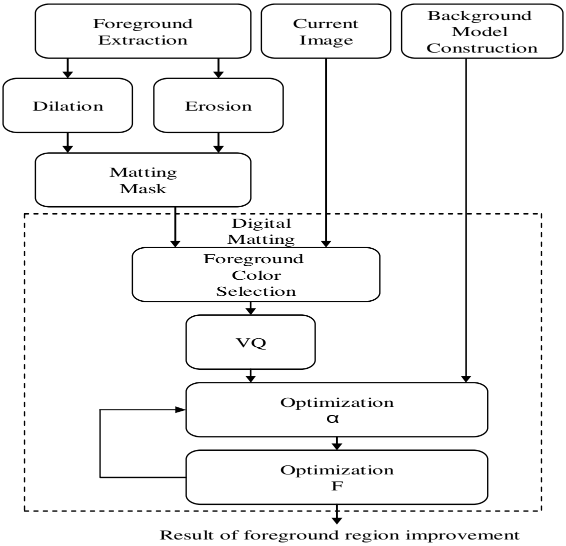


Fig. 6. The flowchart of the digital matting approach

pixels, *F*. Then the dilation operation is applied to original extracted objects to extend the object regions. After subtracting the shrunken object regions from the extended object regions, the "unknown" regions are defined.

Since *F*, *B*, and *C* have RGB channels, individually, we need to solve the problem with three equations and seven unknown parameters. But in our case, the pixel intensity $\mu(x, y)$ of the background model are calculated and known as *B*, so the unknown parameters can be reduced from seven to four. Thus, the problem is formatted as by giving the background color *B* and the observed color *C*, we need to find the best solution for the foreground color *F* and the opacity value α . The optimization process is as following statement. Each pixel in the unknown region is the center of a given window, which is used to define the neighborhood region. So the foreground probability distribution can be built based on the pixel intensities inside this window. The matting problem is then solved by using the maximum a posteriori (MAP) technique to estimate the foreground color *F* and opacity α by giving *B* and *C*. Using the Bayes rule, we can express the result as a sum of log likelihood,

$$\begin{aligned} &\arg\max_{F,\alpha} P(F,\alpha \mid C,B) \\ &= \arg\max_{F,\alpha} P(C,B \mid F,\alpha)P(F)P(\alpha)/P(C)P(B) \\ &= \arg\max_{F,\alpha} L(C,B \mid F,\alpha)+L(F)+L(\alpha) \end{aligned} \tag{5}$$

where *L( )* is log likelihood of *P( )*. We do not care *P(C)* and *P(B)* because they are constant and do not affect the result of the optimization parameters. In addition, we can solve the problem by maximizing the sum of $L(C, B \mid F, \alpha)$, *L(F)*, and $L(\alpha)$.

The first term $L(C, B \mid F, \alpha)$ measures the error between the observed color *C* and the estimate color $\overline{C}$ by estimating *F*, *B*, and α :

$$L(C,B \mid F,\alpha) = -\| C - \alpha F - (1-\alpha)B \|^2 / 2{\sigma_C}^2 \tag{6}$$

The second term *L*(*F*) is used to build the probability distribution of the foreground color, so the *N* (*N*=15 samples) foreground pixels inside the window are selected. In order to be sure of having the robust distribution of the foreground color, each pixel is weighted by:

$$w_n = \max(\| x_n - x_i \|, \| y_n - y_i \|) \tag{7}$$

where (x_i, y_i) is the center location of the window for the pixel *i* in the unknown region, and (x_n, y_n) is the location of the pixel *n* inside this window. Following, the color space has been quantized into several clusters by using VQ (vector quantization) [8]. So each pixel inside the window is classified to corresponding bin in color space. Then we compute the weighted mean $\overline{F}$ and weighted covariance matrix Σ_F of each cluster:

$$\overline{F} = \frac{1}{W} \sum_{n \in N} \frac{1}{w_n} F_n \tag{8}$$

$$\sum\nolimits_F = \frac{1}{W} \sum_{n \in N} \frac{1}{w_n} (F_n - \overline{F})(F_n - \overline{F})^T \tag{9}$$

where $W = \sum_{n \in N} \frac{1}{w_n}$. Subsequently, we use a Gaussian distribution in the RGB color space to model the log likelihoods for foreground *L(F)*:

$$L(F) = -(F - \overline{F})^T \Sigma_F^{-1} (F - \overline{F}) / 2 \tag{10}$$

To find the parameters α and *F*, we assume that *L(* α *)* is constant. First, we assume that *F* values in the RGB color channels are constant and then *C* is projected on the line *FB* to solve α by using following equation:

$$\alpha = (C - B) \cdot (F - B) / \| F - B \|^2 \tag{11}$$

Here, the initial value for *F* is the mean value $\overline{F}$. Second, we assume that α is a constant and then we take the partial derivatives of (5) with respect to *F*. We can find the extreme value when the result of derivatives is equal to 0. Therefore, the best solution for the parameter *F* is obtained.

$$\begin{aligned} &[\ \Sigma_F^{-1} + I\alpha^2 / \sigma_C^2] \quad [F] \\ &= [\ \Sigma_F^{-1} \overline{F} + C\alpha / \sigma_C^2 - B\alpha(1-\alpha) / \sigma_C^2\ ] \end{aligned} \tag{12}$$

where *I* is a 3×3 identity matrix because *F*, *B*, and *C* have three color channels, individually.

The process is computing iteratively to optimize equation (5) until it convergences. Initially, to estimate α by calculating equation (11), we will assume that the *F* values in RGB color channels are fixed, and then to estimate *F* by calculating equation (12), we assume that α is fixed. When we solve equation (5) and optimize the *F* and α, we can get an improvement result of the foreground extraction, as shown in Fig. 7. By comparing the zooming in parts between the original foreground object and the improvement result, we can see that the boundaries of objects are segmented accurately after applying digital matting process.

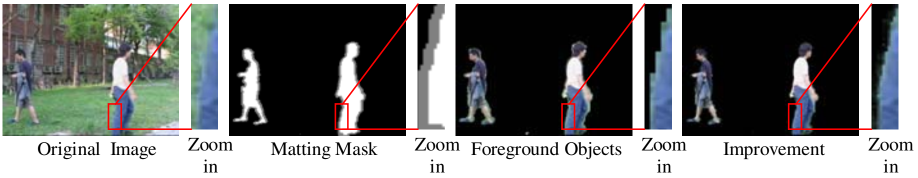


Fig. 7. The improvement result by using digital matting. We can see details by comparing the zooming in parts of the images. After digital matting is applied, we can have visually accurate segmentation result for object's boundaries.

4 Experimental Results

The system proposed in this paper has been successfully tested by several MPEG video sequences. The image size is 320 * 240 pixels. One example is shown in Fig. 8.

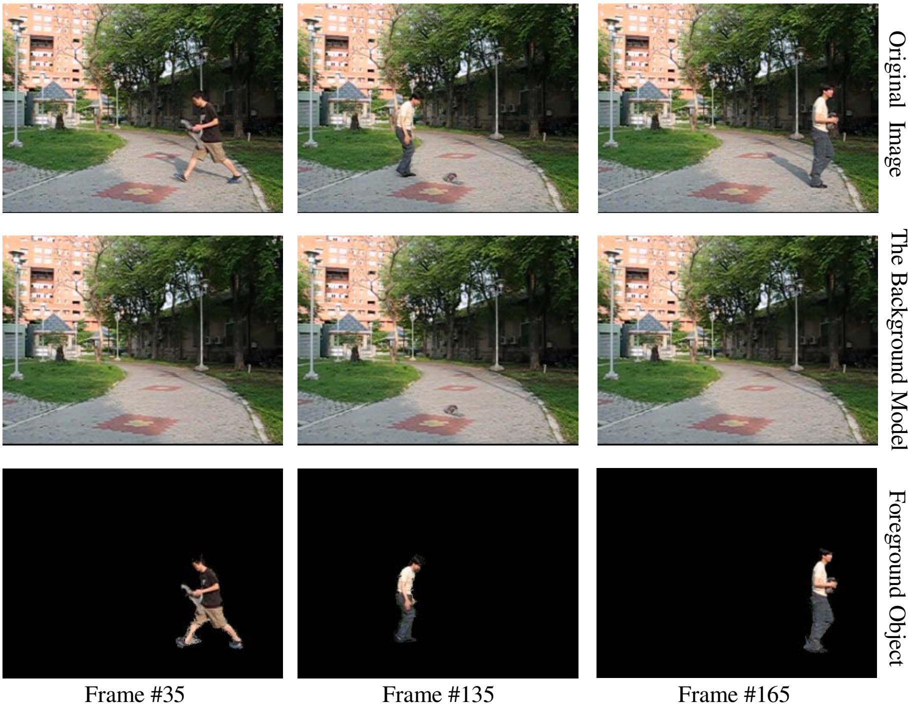
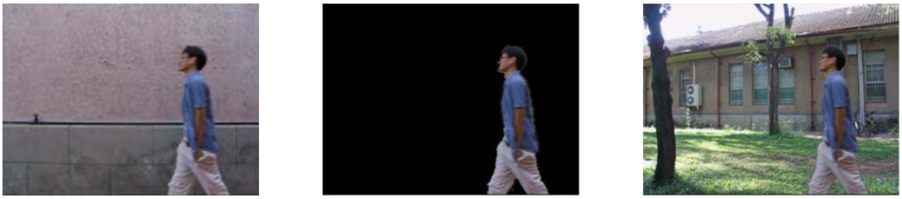


Fig. 8. Segmentation results with updating backgrounds and shadow effects in a video sequence. The deposited object in the image sequence is detected initially. After a period of time, this object would be updated into the background model. In addition, the shadows in the image-sequence are eliminated by using our approach.

In the beginning, the first 60 (M=60) frames of this image sequence are used to construct the background model; meanwhile, there exists a moving object, which is also detected by applying the background subtraction approach during the initial interval of this video sequence. In this outdoor environment, many factors will cause false or missing detections by simply applying the background subtraction process. So the short-term updating approach is applied to quickly updating the background components, which are affected by illumination variances and the periodic movements of the branches or leaves of trees in the background scene. Then, the morphological operations are applied to eliminate the shadows and the remaining false detection components caused by moving branches or leaves of tree. Subsequently, we can find that the deposited jacket is detected initially. But after a period of time, it is updated into the background model by using the long-term updating approach. Finally, the digital matting approach is applied to improve the result of the foreground extraction, especially along the boundaries of the foreground object.

5 Conclusions

In this paper, we developed an automatic and efficient object segmentation system for MPEG video sequences. The system can separate the foreground objects from the background scene by using two modules: the background modeling and updating module and the foreground object extraction module. For the first module, a Gaussian model is applied to build the background model, which is able to be constructed with or without moving foreground objects in the background scene. In the indoor or outdoor environments, background changes all the time caused by the illumination variances or small movements in the background scene. Two updating approaches, the short term and the long term, are adapted to update the background components. For the second module, after the background subtraction approach is applied, the initial foreground regions may have holes due to missing detection for those misclassified pixels, false detections due to small movement components in the background scene and shadows. By using morphological operations, the missing and false detection components can be removed. In addition, the shadow components can be detected and removed by using the linear dependent relation approach. In order to have visually accurate object segmentation, the automatic digital matting approach is applied to improve the result of foreground extraction. Our experimental results show that the performance of this work is promising. Therefore, our system is also capable of applying to the composite image sequences, as shown in Fig. 9.



Original Image Foreground Object Composite Result

Fig. 9. The composite result. By applying the digital matting approach, we can have the composite images by replacing the background scenes in the video sequence.

References

1. Apostoloff, N., Fitzgibbon, A.: Bayesian Video Matting Using Learnt Image Priors. IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, July (2004) I-407-I-414.
2. Chien, S.-Y., Ma S.-Y., Chen, L.-G.: Efficient Moving Object Segmentation Algorithm Using Background Registration Technique. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, July (2002) 577-586.
3. Chien, S.-Y., Ma S.-Y., Chen, L.-G.: Predictive Watershed: A Fast Watershed Algorithm for Video Segmentation. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 5, May (2003) 453-461.
4. Chuang, Y.-Y., Curless, B., Salesin, D. H., Szeliski, R.: A Bayesian Approach to Digital Matting. IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, December (2001) 264-271.
5. Durucan, E., Ebrahimi, T.: Change Detection and Background Extraction by Linear Algebra. Proceeding of the IEEE Volume 89, Issue 10, October (2001) 1368–1381.
6. Elgammal, A., Harwood, D., Davis L.: Non-parametric Model for Background Subtraction. European Conference on Computer Vision, (2000) 751-767.
7. Haritaoglu, I., Harwood, D., Davis L.: W4: Real-Time Surveillance of People and Their Activities. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, August (2000) 809-830.
8. Linde, Y., Buzo, A., Gray, R.: An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications, Vol. Com-28, No. 1, January (1980) 84-95.
9. Suzuki, K., Horiba I., Sugie, N.: Fast Connected-Component Labeling Based on Sequential Local Operations in the Course of Forward Raster Scan Followed by Backward Raster Scan. IEEE International Conference on Pattern Recognition, Vol. 2, (2000) 434 – 437.
10. Zang, O., Klette R.: Robust Background Subtraction and Maintenance. IEEE International Conference on Pattern Recognition, Vol. 2, August (2004) 90-93.

Background Segmentation Beyond RGB

Fredrik Kristensen, Peter Nilsson, and Viktor Öwall

CCCD, Dept. of Electroscience, Box 118, Lund University, SE-221 00 Lund, Sweden
fredrik.kristensen@es.lth.se

Abstract. To efficiently classify and track video objects in a surveillance application, it is essential to reduce the amount of streaming data. One solution is to segment the video into background, i.e. stationary objects, and foreground, i.e. moving objects, and then discard the background. One such motion segmentation algorithm that has proven reliable is the Stauffer and Grimson algorithm. This paper investigates how different color spaces affect the segmentation result in terms of noise and shadow sensitivity. Shadows are especially problematic since they not only distort shape but can also result in falsely connected objects that will complicate tracking and classification. Therefore, a new decision kernel for the segmentation algorithm is presented. This kernel alters the probability of foreground detection to reduce shadows and to increase the chance of correct segmentation for objects with a skin tone color, e.g. faces.

1 Background

Video applications are omnipresent and they are essential for industries such as surveillance, communications, entertainment, and healthcare. A common demand among applications in these fields is that they require large bandwidth and human interaction. To automate the process of separating relevant from irrelevant data, to reduce bandwidth, advanced image processing is necessary, which often requires hardware accelerators to solve the computational bottlenecks. An attempt to automate a surveillance system for indoor monitoring has been initiated, where the goal is to hardware accelerate computational complex parts of a self contained intelligent surveillance camera that can track and classify moving objects.

A conceptual overview of the surveillance system is shown in Fig. 1. The image processing system is supplied with a real-time image stream from a camera, Fig. 1a. A segmentation algorithm preprocesses the image stream and produces a binary mask, Fig. 1b, in which zeros and ones correspond to background and foreground, respectively. In order to remove noise and reconnect split objects, morphologic post processing is performed on the mask [1], Fig. 1c. The object classification part then uses the mask to extract moving parts of the image, Fig. 1d and e, and performs classification and tracking.

This paper addresses two major problems with this type of segmentation, shadows and false-negative detection. Both phenomena are seen in Fig. 1d, where a large shadow is detected to the right of the person in the middle and part of the left person's torso is missing. To reduce these effects, different color spaces and extensions to the decision logic in the segmentation algorithm are investigated.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 602–612, 2006.

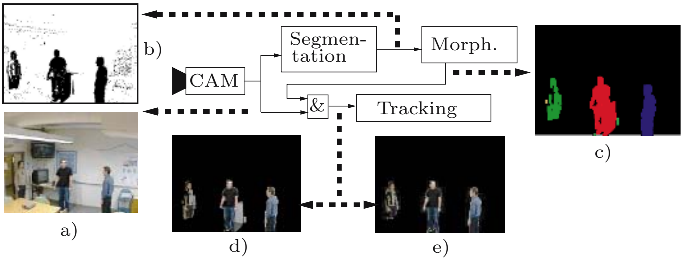


Fig. 1. Surveillance system, (a) original image, (b) binary motion mask, (c) filtered motion mask, (d) detected objects, and (e) detected objects improved result

The goal is to improve performance of the existing segmentation algorithm and not to include additional post processing as in for example [2] and [3].

2 The Segmentation Algorithm

Among many of the algorithms for video segmentation, one based on a Gaussian mixture model [4] was developed with the unique feature of robustness in multi-modal background scenarios and in slowly changing lighting conditions. A multi-modal background distribution is caused by repetitive background object motion, e.g. swaying trees, reflections on a lake surface, flickering of a monitor etc. By representing each pixel process using a mixture of Gaussian distributions, repetitive background motions are merged into one of several background distributions for each pixel.

In short the algorithm works as follows. For each pixel location and distribution a mean, variance, and likelyhood value is stored, based on previous values for this location. A new input is compared to the stored mean values for each distribution at this location. If the difference is less than a constant times the variance, this pixel is part of that distribution and it is updated accordingly. Otherwise a new distribution is created which replaces the least probable distribution. A new input belongs to the background if the distribution that it is part of is one of the most likely distributions. Thus, a foreground pixel that keeps the same color over time is slowly incorporated into the background.

The decision rule, whether a new pixel belongs to an existing distribution or not, can be written as

$$\text{if} \quad (|P_{new} - P^d_{mean}| \leq K P^d_{std}) \quad \text{then} \quad P_{new} \in \text{Distribution } d \tag{1}$$

where P_{new} is the new pixel value, P^d_{mean} and P^d_{std} are the stored mean value and standard deviation of distribution d, and K is a constant. In the original paper [4], K is set to a fixed value of 2.5 based on experimental results. In this paper we suggest to change K for two different cases. First, if P_{new} could be part of a moving shadow, K is increased, i.e. the sensitivity is decreased, and second, if P_{new} has skin tone color, K is decreased. These changes are discussed further in Section 3.

To reduce computational complexity the algorithm assumes that the three input colors of each pixel are independent. This avoids costly matrix inversions. However, all color spaces consists of more or less dependent color channels, which will reduce accuracy of the algorithm. The used color space will also determine how sensitive the algorithm is to shadows. Therefore, different color spaces have been investigated with regard to both shadow and noise sensitivity.

3 Color Spaces and Shadows

3.1 Shadows

Shadows can be divided into two classes, dynamic and static shadows. Dynamic shadows occur when an object moves in between a light source and the background or another object, and static shadows are cast by static objects in the scene. Segmentation algorithms based on statistical background models, as the one used in this paper, are not affected by static shadows, since they are incorporated as part of the background model. However, dynamic shadows are of major interest since they can be erroneously detected as an object.

In order to remove shadows, pixels that could be part of a shadow have to be identified. In the RGB space, difference in color between a pixel before and after it becomes part of a shadow can be simplified to

$$C_{shadow} = \alpha_c C_{light}\ C, c \in \{R, G, B\},\ \alpha_c \leq 1, \tag{2}$$

where α_c depends on the present light sources and reflection terms of the surfaces [5]. For example, with a white light and uniform reflection terms, α_c is equal for all three colors. White light means that the color spectrum is flat, i.e. all colors are equally represented. In addition to light sources and reflection terms the used color space will also affect the behavioral of a shadow.

3.2 Color Spaces

Many different color space models can be found in the literature, but the most commonly used are RGB, HSI, and YC$_b$C$_r$. RGB is the color space commonly acquired directly from a sensor or camera, since a color sensor often uses red, green, and blue filters to obtain color information. Red, Green, and Blue are usually measured with 8-bits resolution, where 0 is no color at all and 255 is the maximum color, hence the 24-bit true color definition. HSI and YC$_b$C$_r$ are closer to human interpretation of colors in the sense that brightness, or intensity, is separated from the base color. YC$_b$C$_r$ uses Cartesian coordinates to describe the base color while HSI uses polar coordinates. In addition to these three color spaces, normalized-RGB (rgb), $C_1C_2C_3$, $l_1l_2l_3$, and $m_1m_2m_3$ are investigated [6]. They are all invariant to changes in brightness, a feature that should decrease their sensitive to shadows. Equation 3 to 8 show the relationship between RGB and respective color space, since it is assumed that the camera or sensor output is in the RGB color space.

$$H = \begin{cases} \theta & \text{if } B \leq G, \quad \theta = \arccos\left\{\frac{0.5[2R-G-B]}{[(R-G)^2+(R-B)(G-B)]^{0.5}}\right\} \\ 360-\theta \text{ if } B > G \end{cases} \tag{3}$$

$$S = 1 - \frac{3[\min(R,G,B)]}{R+G+B}, \quad I = \frac{1}{3}(R+G+B)$$

$$rgb = \tfrac{C}{R+G+B}, \, C \in \{R,G,B\} \tag{4}$$

$$Y = 0.257R + 0.504G + 0.098B + 16 \tag{5}$$

$$C_b = -0.148R - 0.291G + 0.439B + 128, \quad C_r = 0.439R - 0.368G - 0.071B + 128$$

$$C_1C_2C_3 = \tan^{-1}\left(\frac{R}{\max(G,B)}\right), \tan^{-1}\left(\frac{G}{\max(R,B)}\right), \tan^{-1}\left(\frac{B}{\max(R,G)}\right) \tag{6}$$

$$l_1l_2l_3 = \frac{(C)^2}{(R-G)^2+(R-B)^2+(G-B)^2}, \; C \in \{R-G, R-B, G-B\} \tag{7}$$

$$m_1 = \frac{R_1G_2}{R_2G_1} \quad m_2 = \frac{R_1B_2}{R_2B_1} \quad m_3 = \frac{B_1G_2}{B_2G_1}, \tag{8}$$

where RGB_1 and RGB_2 are two neighboring pixels.

To compare color spaces in terms of noise and shadow sensitivity, two properties are investigated. First, how the color space handles small changes in RGB, since these should not cause a pixel to accidentally become foreground. Small changes in RGB are common in a video sequence due to noise and preprocessing in the camera. Secondly, how the color space reacts to changes in lighting conditions. With a high sensitivity to light changes, shadows will be detected, while on the other hand low sensitivity to light could result in missed object detections.

An example on how the different color spaces react to noise and lighting changes are presented in Table 1, where $m_1m_2m_3$ is excluded since it depends on pixel pairs. The noise property is tested both for a dark color, i.e. RGB close

Table 1. Noise and light properties. For each property and color space, two triplets are presented, the top one shows the original values and the lower one shows the change in color when affected with noise or brightness changes. In RGB, noise is a small change in each color channel, and for decreased and increased brightness each color channel is multiplied with 0.7 and 1.4, respectively.

Mode	RGB	HSI	YC$_b$C$_r$	rgb	$C_1C_2C_3$	$l_1l_2l_3$
noise in a dark color	(1,3,2) (1, -2, 1)	(107,171,2) (85, 0, 0)	(18,128,128) (1, 1, 1)	(43,128,85) (42,-85,43)	(52,159,95) (43,-107,64)	(171,43,43) (-128,0,128)
noise in a bright color	(180,129,172) (2,-2,-2)	(220,73,160) (3,4,0)	(144,140,148) (-1,-1,2)	(96,69,92) (1,-1,-1)	(131,101,124) (2,-2,-2)	(148,4,105) (6,3,-11)
decreased brightness	(90,130,150) (-27,-39,-45)	(142,102,123) (0,0,-37)	(119,143,110) (-31,-4,5)	(62,90,104) (0,0,0)	(88,116,139) (0,0,0)	(73,165,18) (0,0,0)
increased brightness	(130,160,90) (52,64,36)	(61,112,127) (0,0,50)	(139,102,120) (49,-10,-3)	(88,108,61) (0,0,0)	(111,144,83) (0,0,0)	(31,55,170) (0,0,0)

to zero, and for a bright color. The light property is tested both for a decrease and an increase in brightness. White light is assumed and all colors have been fitted to an 8-bit representation, i.e. 0-255.

In Table 1 it is seen that HSI, rgb, $C_1C_2C_3$, and $l_1l_2l_3$ are highly sensitive to noise in the dark color, $l_1l_2l_3$ is in addition also somewhat sensitive to noise in the bright color. The only color spaces that handle noise well are RGB and YC$_b$C$_r$. HSI is sensitive to noise due to its polar coordinate description. For colors close to the origin, i.e. the gray scale, a small change can result in a large change of the H component. In Equation 7 it is seen that $l_1l_2l_3$ is also sensitive to changes close to the gray scale, since it is normalized with the differences between R, G, and B. The color spaces rgb, $C_1C_2C_3$, and $m_1m_2m_3$ are all normalized, see Equation 4, 6, and 8, which means that they are unstable when RGB is close to zero.

The effect of a change in brightness differs much between color spaces. In RGB all three channels change values, while for HSI only the I channel changes. YC$_b$C$_r$ has the largest change in the Y component and only minor changes in C$_b$ and C$_r$. Remaining color spaces, rgb, $C_1C_2C_3$, and $l_1l_2l_3$, are affected neither by an increase nor a decrease in brightness since they are normalized, i.e. they are light invariant.

3.3 Exprimental Results

To experimentally evaluate the segmentation algorithm a short video sequence of a walking human was recorded and transformed to all different color spaces. All colors have been fitted to an 8-bit representation and in the case of $m_1m_2m_3$, RGB$_1$ and RGB$_2$ are taken to be two horizontally adjacent pixels. The background in the video is mostly white and gray but many static objects are included with different properties, e.g. shiny, colorful, and matt. Three different light sources are present, overhead fluorescent light, day light from the windows to the right of the scene, and a strong spotlight (not visible) to introduce additional shadows. Fig 2 shows the segmentation results of each color space for a video frame in the middle of the sequence. The segmentation result is presented as the input image with the background set to black.

Many of the experimental results confirm theory. HSI, rgb, $C_1C_2C_3$, $l_1l_2l_3$, and $m_1m_2m_3$ are noisy but less sensitive to shadows than RGB and YC$_b$C$_r$. Additional observations are:

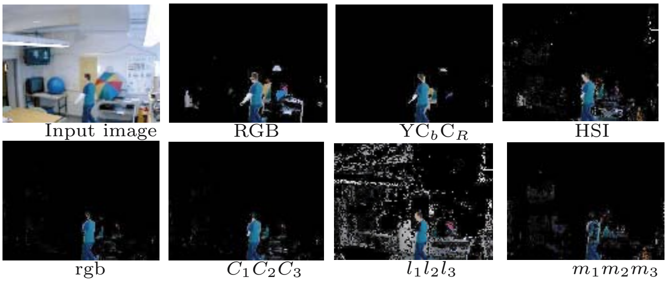


Fig. 2. Input image and segmentation result for the different color spaces

1. $\mathrm{YC}_b\mathrm{C}_r$ experiences less noise than RGB, due to the more independent color channels. The segmentation algorithm assumes that all color channels are independent, see Section 2.
2. Even though both HSI and $l_1l_2l_3$ are sensitive to changes close to the gray scale, the result is much worse for $l_1l_2l_3$.
3. The light invariant color spaces do not detect as much shadows as the other color spaces, at a cost of missed detection of bright areas. This is easiest seen on the person's white arm in Fig. 2.
4. With the $m_1m_2m_3$ color space the segmentation algorithm becomes more of an edge detector, since it is based on two neighboring pixels.

The overall most suitable color space for the segmentation algorithm is $\mathrm{YC}_b\mathrm{C}_r$. It is least sensitive to noise, due to numerical stability and more independent color channels. No information is lost when it is calculated from RGB compared to the normalized color spaces in which brightness information is lost. However, it is affected by shadows and compared to RGB it is too insensitive in some cases, e.g. compare the face detection of RGB and $\mathrm{YC}_b\mathrm{C}_r$ in Fig. 2b and c. Section 4 presents compensation methods for these two cases.

4 Performance Improvement

4.1 Increased Sensitivity

Note in Equation 5 that the complete dynamic range is not utilized. Maximum and minimum values are $\{235, 240, 240\}$ and $\{16, 16, 16\}$ respectively. This is due to the intended original application of $\mathrm{YC}_b\mathrm{C}_r$, television broadcast [7]. However, in our application no extra information has to be transmitted and the dynamic range can be extended to 0-255.

In $\mathrm{YC}_b\mathrm{C}_r$ the Y component contains more information than the color components, as shown in Fig. 3a. This is also observed in the segmentation result shown in Fig. 3b. Two observations can be made from this; first, C_b and C_r have low sensitivity to shadows and secondly, the color variance is lower than gray scale variance. To increase the amount of information in C_b and C_r the following transformation from RGB is proposed:

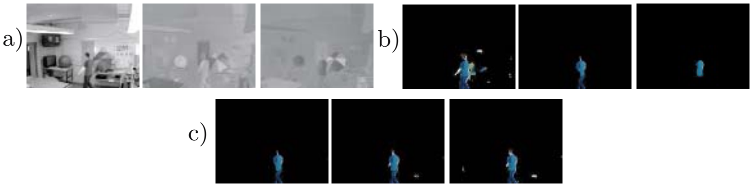


Fig. 3. a) Original image divided into the three color channels, Y, C_b, and C_r. b) Segmentation result for each color channel with q=1. c) Segmentation result for $\mathrm{C}_b\mathrm{C}_r$ with $q = 1$, $q = 1.3$, and $q = 1.6$.

$$Y = 0.299R + 0.586G + 0.114B \tag{9}$$

$$X_b = (-0.167R - 0.330G + 0.497B)q + 128 \quad X_r = (0.497R - 0.417G - 0.080B)q + 128,$$

$$C_b = \begin{cases} 255 & \text{if } X_b > 255 \\ 0 & \text{if } X_b < 0 \\ X_b & \text{else} \end{cases} \qquad C_r = \begin{cases} 255 & \text{if } X_r > 255 \\ 0 & \text{if } X_r < 0 \\ X_r & \text{else} \end{cases}$$

where q is a scale factor. With $q = 1$ the range is from 0-255 and no over- or underflow will occur. If $q > 1$ the sensitivity for mid range colors are increased at the expense of decreased sensitivity for pure colors. An example is shown in Fig. 3c, where the segmentation results for $\mathrm{C}_b\mathrm{C}_r$ with q =1, 1.3, and 1.6 are shown. As q is increased, more and more of the person is detected and more noise is introduced.

The optimal value for q depends on data and has to be adjusted from case to case and over time as lightning conditions and background change. To change q during runtime is possible as long as the change is slow, since any sudden change in color will result in foreground detection. To automate this process two measurements can be used; Number of under- and overflows in $\mathrm{C}_b\mathrm{C}_r$ and the number of small isolated objects. For example in Fig. 3c, one such noise object can be seen in the middle picture and two in the right picture.

4.2 Shadow Reduction

To reduce the number of detected shadows, pixels that could be part of a potential shadow have to be recognized. Using Equation 2 and 9 it is seen that Y will always be smaller when shaded and that C_b and C_r will go towards 128, i.e. the origin. Since Y is calculated as a sum of the RGB colors, changes due to shadows will generally be larger in Y than in $\mathrm{C}_b\mathrm{C}_r$ which are calculated from the differences between R, G, and B. With this information a simple rule for shadows can be formed; A potential shadow is found if a large negative change is detected in Y and $\mathrm{C}_b\mathrm{C}_r$ have moved slightly towards origin, compared to the stored mean of $\mathrm{YC}_b\mathrm{C}_r$. However, if noise is taken into consideration this rule might no longer hold, C_b or C_r could actually move away from the origin when shaded. This means that a small error margin has to be incorporated in the rule.

In the upper part of Fig. 4 three different rules for the $\mathrm{C}_b\mathrm{C}_r$ plane to detect potential shadows are shown. All three assume that a negative change in Y has already been detected. The gray areas represent the part of the $\mathrm{C}_b\mathrm{C}_r$ plane where a new pixel is ruled to be a potential shadow. The area location is based on the stored mean of C_b and C_r ($\overline{C_b}\,\overline{C_r}$), five example points are shown for each rule. The first rule, shown in Fig. 4b, is the simplest rule. A new pixel is part of a potential shadow if the sum of absolute differences in $\mathrm{C}_b\mathrm{C}_r$ is less than a threshold. The second rule, shown in Fig. 4c, compares the difference in C_b and C_r separately to thresholds that depends on the position of $\overline{C_b}\,\overline{C_r}$. The last rule, shown in Fig. 4d, allows changes in a small sector from origin or, if the values are close to origin, in a small box around $\overline{C_b}\,\overline{C_r}$ [8]. This rule is the most complex rule, since it involves a division to approximate the arctan function.

The segmentation result with the three different shadow rules are shown in Fig. 4. How the rules are used in the segmentation algorithm are explained in

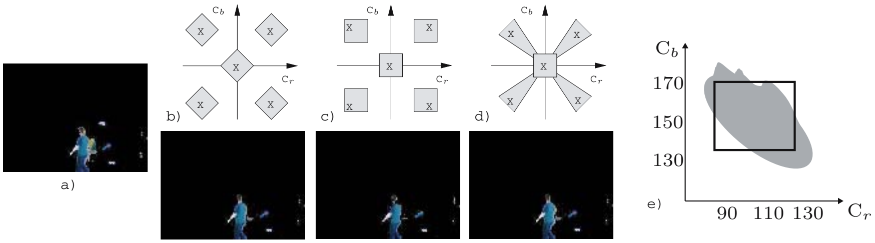


Fig. 4. The original result (a) and the result with three different shadow detection rules (b,c,d) in the C_bC_r plane, where X is the stored mean of C_bC_r, five example points are shown for each rule. A new pixel part of a potential shadow if it falls in the gray area and there is a negative change in the Y component. (e) Skin color distribution in the C_bC_r plane and the thresholds used to find skin color.

Section 4.4. As seen, most of the shadows are removed compared to Fig. 4a, with all three rules. However, the rule in Fig. 4d removes some parts of the object as well, due to acceptance of large changes in color and that color can move away from origin and still be classified as a shadow. In this particular image no significant performance difference can be seen between the two simpler rules.

4.3 Skin Tone

If any form of human recognition is to be included in the surveillance system, face detection becomes very important. To increase the chances of correct segmentation of faces, we try to find pixels with skin tone and use that information to improve the foreground/background decision. Skin tones differ much for human races, from black to white and with different tones of red and yellow. However, in the YC_bC_r color space, these colors are tightly distributed in the C_bC_r plane along the Y axis. This means that the Y component can be disregarded, since human skin tone has about the same color distribution in C_bC_r for most Y values [9][10][11]. In Fig. 4e, a simplified reprint of the C_bC_r distribution found in [9] together with the thresholds used in this paper to find skin color are shown. The distribution applies to faces with a Y component in between 60-175. Thresholds are chosen for simplicity and not for perfect matching, since the result will only be used to increase the chance of correct segmentation.

4.4 Extended Segmentation Logic

With the methods described previously in Section 4 pixels that are likely to be part of a shadow or human skin can be found. With this information the decision kernel of the segmentation algorithm, see Equation 1, can be altered to vary the likelihood of including these pixels as part of the foreground. Shadows should have a lower and human skin should have a higher probability to be included in the foreground. Below a more advanced decision kernel is proposed. This kernel is executed once for each stored distribution or until a matching distribution is found.

```
/* Advanced decision kernel */
 skin = 1;  shadow = 1; Threshold = 2.5 * Standard_deviation;
 d’Y = Y_new - Y_mean; d’Cb = Cb_new - Cb_mean; d’Cr = Cr_new - Cr_mean;

 if (d’Y<0 and d’Y>-Max_shadow and d’CbCr<Shadow_rule) then shadow=2;
 if (CbCr == skin color ) then skin=0.5;

 distribution_found = true;
 if (d’Y>skin*Threshold or d’Y<-skin*shadow*Threshold)
     distribution_found = false;
 if (abs(d’Cb) > skin*Threshold)
     distribution_found = false;
 if (abs(d’Cr) > skin*Threshold)
     distribution_found = false;
```

where *shadow_rule* is one of the rules described in Section 4.2 and $Max\_sha-dow$ express how dark a shadow is allowed to be. For example, if *Max_shadow* is set to 255 a pixel that change color from pure white to pure black would be classified as a shadow. Most indoor scenarios do not have such dark shadows and this result in a loss of sensitivity instead of a shadow reduction. Based on experimental results, a suitable indoor value for *Max_shadow* is around 50-80.

In the modified kernel the threshold value to find a matching distribution is based on the standard deviation and it is increased if a potential shadow is detected. This method, compared to take a hard decision about shadows, allows the algorithm to exploit the property of the standard deviation which generally is higher in bright areas and lower in dark areas [4]. Thus, a higher threshold is used for a shaded bright area than for a shaded dark area. A non-uniform decision threshold reduces the number of misclassified pixels, since a higher threshold is required to avoid foreground detection of a bright area that becomes shaded compared to a dark area that becomes shaded. With a uniform threshold the segmentation algorithm would be too insensitive in dark areas or classify shadows as foreground in bright areas.

There are two effects of increased sensitivity in skin colored areas. First, any visible skin is more probable to be detected as foreground, which is desired. Secondly, background areas that are skin colored produce more random foreground noise. However, most noise is removed by the morphologic filtering that follows the segmentation, as shown in Fig 1. With this approach it is not crucial to choose perfect decision boundaries for skin colors. Objects with naked skin that are not classified as skin are not automatically mistaken as background, they will in most cases be correctly classified as foreground by the unaltered decision rule. Hence, a low complexity skin color threshold, like the one shown in Fig. 4e, can be used.

5 Results

In Fig. 5 the step-by-step improvements outlined in Section 4 are shown, from original RGB segmentation to the morphologic filtered YC_bC_r segmentation with stretched color space, suppressed shadows, and increased skin color sensitivity. The two largest noise objects to the right in Fig. 5f are due to increased lightning, since the person in the video entered from the right and blocked the window light

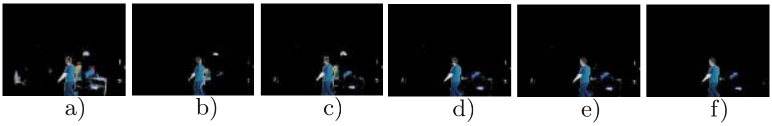


Fig. 5. Segmentation result with: (a) RGB, (b) YC_bC_r, (c) stretched YC_bC_r, (d) shadow suppression, (e) increased skin sensitivity, and (f) morphologic filtered. All effects in (b)-(f) are cumulative.

before entering. Since these objects are not shadows they are correctly detected as foreground, but they will disappear with time as the background model adapts.

All results presented in this paper are retrieved from the same video frame, for comparison reasons. However, all results are data dependent and must be verified for a large set of input data with varying backgrounds and lighting conditions. To simulate this is a very time consuming process. Thus, only a limited set of scenarios have been tested so far, all with a promising degree of success. For example, Fig. 1e shows the improved result of the original example. Video results are found on the homepage [12]. For the same reasons, time and a limited test set, no exact numbers on the input parameters are presented. Extensive testing will begin as soon as the first part of the system, i.e. the sensor, segmentation algorithm, and morphologic filter, is integrated on an FPGA board. These hardware architectures are presented in [13] and [14]. Reduced simulation time means that long-term effects and fine-grain adjustment of parameter settings will be possible to investigated.

6 Conclusions

In this paper it is shown how different color spaces affect the result of a Gaussian mixture model segmentation algorithm. The YC_bC_r is found to be best in terms of noise, due to numeric stability and an independent brightness channel. Since the YC_bC_r color space is sensitive to shadows, three different classification rules to identify shadows are investigated and it is found that the simplest rule is as good as the more advanced.

A more advanced decision kernel for the segmentation algorithm is presented. With this kernel the probability to detect a foreground pixel is altered depending on the nature of the pixel. If a pixel is part of a potential shadow the probability is reduced and if the pixel has a human skin tone the probability is increased. It is found that with these small additions to the decision kernel the segmentation result is significantly improved without costly post processing.

References

1. Goutsias, J., Heijmans, H.J.: Fundamenta Morphologicae Mathematicae. Fundamenta Informaticae **41** (2000) 1–31
2. Nadimi, S., Bhanu, B.: Moving shadow detection using a physics-based approach. In: Proc. of International Conference on Pattern Recognition (ICPR 2002), Quebec, Canada (2002)

3. Xu, D., Li, X., Liu, Z., Yuan, Y.: Cast shadow detection in video segmentation. Pattern Recognition Letters **26** (2005) 91–99
4. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), Ft. Collins, CO, USA (1999)
5. Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast shadow segmentation using invariant color features. Computer Vision and Image Understanding **95** (2004) 238–259
6. Gevers, T., Smeulders, A.: Color-based object recognition. Pattern recognition, The Journal of the pattern recognition society **32** (1999) 453–464
7. International Telecommunication Union: ITU-R BT.601, Studio encoding parameters of digital television (1987) www.itu.int/ITU-R/.
8. Schreer, O., Feldmann, I., Gölz, U., Kauff, P.: Fast and Robust Shadow Detection in Videoconference Apllication. In: 4th EURASIP-IEEE Region 8 Int. Symposium on Video/Image Processing and Multimedia Communications, Zadar, Croatia (2002)
9. Wong, K., Lam, K., Siu, W.: An Efficient Color Compensation Scheme for Skin Color Segmentation. In: Proc. of IEEE International Symposium on Circuits and Systems (ISCAS'03), Bangkok, Thailand (2003)
10. Garcia, C., Tziritas, G.: Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis. IEEE Trans. Multimedia **1** (1999) 264–277
11. Wang, H., Shang, S.: A Highly Efficient System for Automatic Face Region Detection in MPEG Video. IEEE Trans. Circuits Syst. Video Technol. **7** (1997) 615–628
12. : Project homepage (2005) www.es.lth.se/home/fkn/index.html.
13. Jiang, H., Ardö, H., Öwall, V.: Hardware accelerator design for video segmentation with multi-modal background modelling. In: Proc. of IEEE International Symposium on Circuits and Systems (ISCAS'05), Kobe, Japan (2005)
14. Hedberg, H., Kristensen, F., Nilsson, P., Öwall, V.: A low complexity architecture for binary image erosion and dilation structuring element decomposition. In: Proc. of IEEE International Symposium on Circuits and Systems (ISCAS'05), Kobe, Japan (2005)

Classification of Photometric Factors Based on Photometric Linearization

Yasuhiro Mukaigawa[1,*], Yasunori Ishii[2,*], and Takeshi Shakunaga[3]

[1] The Institute of Scientific and Industrial Research, Osaka University,
8-1 Mihogaoka, Ibaraki-shi, Osaka 567-0047, Japan
mukaigaw@am.sanken.osaka-u.ac.jp
[2] Matsushita Electric Industrial Co., Ltd.
[3] Department of Computer Science, Okayama University,
Okayama-shi, Okayama 700-8530, Japan

Abstract. We propose a new method for classification of photometric factors, such as diffuse reflection, specular reflection, attached shadow, and cast shadow. For analyzing real images, we utilize the photometric linearization method which was originally proposed for image synthesis. First, we show that each pixel can be photometrically classified by the simple comparison of the pixel intensity. Our classification algorithm requires neither 3D shape information nor color information of the scene. Then, we show that the accuracy of the photometric linearization can be improved by introducing a new classification-based criterion to the linearization process. Experimental results show that photometric factors can be correctly classified without any special device.

1 Introduction

The appearance of an object changes due to lighting direction and surface reflectance. Since real images include complex factors such as specular reflections and shadows, it is difficult to directly apply various computer vision algorithms, such as photometric stereo[1], to real images. Therefore, it is important to analyze the photometric factors included in real images.

A lot of methods have already been proposed for separating photometric factors. The dichromatic reflection model [2] is often used for separating diffuse reflections and specular reflections [3, 4, 5]. Wolff et al.[6] proposed a method to separate specular reflections by analysis of reflected polarization, Nayar et al.[7] combined color and polarization to separate specular reflections. Ikeuchi et al.[8] proposed a method to classify photometric factors based on range and brightness images. These methods, however, have a common restriction in that shadows cannot be analyzed.

On the other hand, there are some methods which express real images in a linear subspace. Shashua[9] showed that an image lighted from any direction can be expressed by a linear combination of three base images taken under different lighting directions under the assumption of a Lambertian surface and a parallel

[*] This work was mainly accomplished when the authors were with Okayama University.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 613–622, 2006.

ray. That is, an image can be perfectly expressed in a 3-D subspace. Belhumeur and Kriegman[10] showed that an image can be expressed by the illumination cone model even if the image includes attached shadows. In the illumination cone, images are expressed by a linear combination of extreme rays. Georghiades et al.[11] developed the illumination cone so that cast shadows can be also expressed by the shape reconstruction. Although any photometric factors can be ideally expressed by the illumination cone, a large number of images corresponding to extreme rays are necessary.

We have proposed the photometric linearization method[12], which converts real images into ideal images that include only diffuse factor. After the photometric linearization, all images are expressed as a linear combination of three base images. The method was originally proposed for image synthesis. In this paper, we show that the method can also be used for classifying photometric factors. It can classify not only diffuse reflections and specular reflections, but also attached shadows and cast shadows. We present a new criterion for classification of photometric factors based on the photometric linearization. The classification algorithm requires neither 3D shape information nor color information of the scene. The classification is accomplished by the simple comparison of pixel intensities.

Moreover, we show that the accuracy of the original photometric linearization can be improved by introducing a new classification-based criterion to the linearization process. The original photometric linearization method does not work stably when pixels are not illuminated in a number of input images. Our physics-based analysis can solve this problem.

2 Classification

2.1 Photometric Factors

Photometric factors are classified into reflections and shadows (Fig.1). The reflections are classified into diffuse reflections and specular reflections. According to the Lambert model, the intensity of the diffuse reflection is expressed by

$$i = \boldsymbol{n}^T \boldsymbol{s}. \tag{1}$$

Here, $\boldsymbol{n}$ denotes the surface property vector which is a product of the unit normal vector and the diffuse reflectance, and $\boldsymbol{s}$ denotes the lighting property vector which is a product of the unit vector along the lighting direction and the

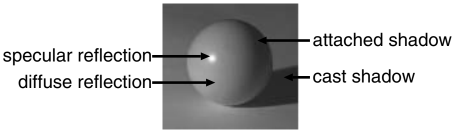


Fig. 1. Photometric factors included in an image

lighting power. The specular reflections are observed as the sum of diffuse factors and specular factors.

Shadows are classified into attached shadows and cast shadows. Attached shadows depend on the angle between the surface normal and the lighting direction and are observed where the surface does not face the light source. Cast shadows depend on the overall 3-D shape of the scene, and are observed where light is occluded by other objects. If there is no ambient light and interreflection, the intensity in shadows becomes zero. However, Eq.(1) indicates that the intensity in attached shadow is negative, while that in cast shadow is positive.

2.2 Photometric Linearization

We have proposed the photometric linearization method[12] which converts real images including various photometric factors into ideal images including only diffuse reflection factor. After the photometric linearization, all pixels in images fully satisfy Eq.(1). Hence, any image can be expressed by a linear combination of three base images[9].

For the photometric linearization, multiple images are taken under various lighting directions. The camera and target objects are fixed. It is important that the lighting direction, the 3-D shape of the target objects, and the reflectance of the surface are unknown.

2.3 Criterion for Classification

In this section, we show that each pixel can be easily classified into diffuse reflection, specular reflection, attached shadow, and cast shadow based on the photometric linearization. The classification is accomplished by the simple comparison of the pixel intensity.

Let $i_{(k,p)}$ be the intensity of the pixel p in the image k, and let $i^L_{(k,p)}$ be the linearized intensity. The relationship between $i_{(k,p)}$ and $i^L_{(k,p)}$ is as follows. In the diffuse reflection region, $i^L_{(k,p)}$ is equal to $i_{(k,p)}$, because the intensity is not changed by the linearization. In the specular reflection region, $i^L_{(k,p)}$ is smaller than $i_{(k,p)}$, because the specular factor is eliminated. In the attached shadow region, $i^L_{(k,p)}$ becomes negative, which satisfies Eq.(1). In the cast shadow region, $i^L_{(k,p)}$ is larger than $i_{(k,p)}$, because $i^L_{(k,p)}$ has a diffuse reflection factor while $i_{(k,p)}$ is near zero. Hence, each pixel can be classified by the following criterion:

$$Region(k,p) = \begin{cases} D : if\ (|i_{(k,p)} - i^L_{(k,p)}| \leq T \times i_{(k,p)})\ \cap\ (i_{(k,p)} \geq T_s) \\ S : if\ (i_{(k,p)} - i^L_{(k,p)} > T \times i_{(k,p)})\ \cap\ (i^L_{(k,p)} \geq 0)\ \cap\ (i_{(k,p)} \geq T_s) \\ A : if\ (i^L_{(k,p)} < 0)\ \cap\ (i_{(k,p)} < T_s) \\ C : if\ (i^L_{(k,p)} \geq 0)\ \cap\ (i_{(k,p)} < T_s) \\ U : otherwise \end{cases} \tag{2}$$

Here, D,S,A,C, and U denote diffuse reflection, specular reflection, attached shadow, cast shadow, and undefined factor, respectively. The threshold T is

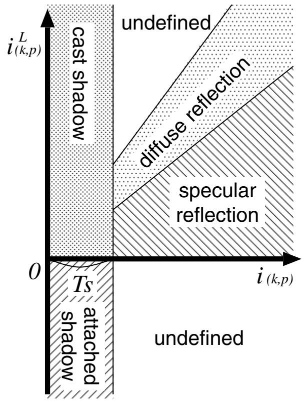
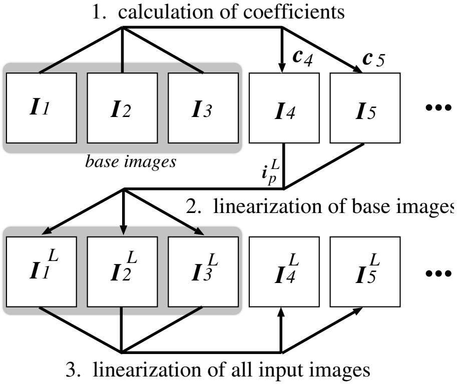


Fig. 2. Criterion for classification of photometric factors



1. calculation of coefficients
c_4
c_5
I_1
I_2
I_3
I_4
I_5
•••
base images
i^L_p
2. linearization of base images
I^L_1
I^L_2
I^L_3
I^L_4
I^L_5
•••
3. linearization of all input images

Fig. 3. Flow of the linearization process

used to check the equality of $i_{(k,p)}$ and $i^L_{(k,p)}$, and empirically determined. Since T is normalized to be relative to $i_{(k,p)}$, the check becomes independent of the brightness. In real images, the intensities of shadows are not zero. The threshold T_s is used to distinguish shadows, and can be determined by manually sampling some pixels in shadow regions.

In this criterion, the shadow regions are classified just by using threshold T_s. Although the classification is very simple, attached shadows and cast shadows can be distinguished by the sign of $i^L_{(k,p)}$. It is one of the significant advantages of the criterion because two types of shadows can be distinguished without any 3D shape information. Figure 2 illustrates Eq.(2) as a 2-D plane spanned by $i_{(k,p)}$ and $i^L_{(k,p)}$. The photometric factors are easily classified if the photometric linearization is accomplished.

3 Improvement of Photometric Linearization

3.1 Key Idea

In the previous section, we showed that photometric factors are correctly classified if the photometric linearization is perfectly accomplished. That is, any pixel is never classified into the undefined factor. This fact suggests that the photometric linearization becomes more accurate by introducing the criterion for classification to the linearization process. We can use the criterion to verify the accuracy of the photometric linearization.

3.2 Flow of the Process

First, we summarize the photometric linearization. Shashua[9] showed that if a parallel ray is assumed, an image $\boldsymbol{I}_k$ under any lighting direction can be expressed by a linear combination of three base images ($\boldsymbol{I}_1$, $\boldsymbol{I}_2$, and $\boldsymbol{I}_3$) taken under different lighting directions,

$$\boldsymbol{I}_k = c^1_k \boldsymbol{I}_1 + c^2_k \boldsymbol{I}_2 + c^3_k \boldsymbol{I}_3. \tag{3}$$

Here, let $\boldsymbol{c}_k = [\ c_k^1\ c_k^2\ c_k^3\ ]^T$ be a set of coefficients of the image $\boldsymbol{I}_k$. Real images, however, do not satisfy Eq.(3), because shadows and specular reflections are observed. The photometric linearization can convert real images to ideal images which perfectly satisfy Eq.(3). The process of the photometric linearization is divided into the following three steps (Fig.3).

1. **Calculation of a set of coefficients**
   First, three base images $\boldsymbol{I}_1$, $\boldsymbol{I}_2$, and $\boldsymbol{I}_3$ are selected from among the input images. A set of coefficients $\boldsymbol{c}_k$ of the k-th input image $\boldsymbol{I}_k$ is calculated from the intensities in $\boldsymbol{I}_1$, $\boldsymbol{I}_2$, $\boldsymbol{I}_3$, and $\boldsymbol{I}_k$.
2. **Photometric linearization of base images**
   Next, the base images are linearized for every pixel based on the input images and the coefficients. Let $\boldsymbol{i}_p^L = [\ i_{(1,p)}^L\ i_{(2,p)}^L\ i_{(3,p)}^L]^T$ be a set of intensities in the linearized base images at pixel p. This process is performed for all pixels, and three base images $\boldsymbol{I}_1$, $\boldsymbol{I}_2$, and $\boldsymbol{I}_3$ are converted into the linearized base images $\boldsymbol{I}_1^L$, $\boldsymbol{I}_2^L$, and $\boldsymbol{I}_3^L$.
3. **Photometric linearization of all images**
   Finally, all input images are linearized. The k-th input image $\boldsymbol{I}_k$ is linearized by the linear combination of the linearized base images $\boldsymbol{I}_1^L$, $\boldsymbol{I}_2^L$, and $\boldsymbol{I}_3^L$ using $\boldsymbol{c}_k$. We denote the linearized $\boldsymbol{I}_k$ as $\boldsymbol{I}_k^L$.

3.3 Calculation of Candidates by Random Sampling

The coefficients of the linear combination and the base images have to be determined to satisfy Eq.(3). If we calculate them by minimizing root mean square errors, input images are not converted to ideal images that include only diffuse factor because of shadows and specular reflections.

The photometric linearization solves this problem by the RANSAC-based approach. A lot of candidates are iteratively calculated by random sampling, and the correct value calculated from only diffuse reflections is selected from among the candidates. If all pixels are sampled from the diffuse reflection region, the correct value, which is not affected by specular reflections and shadows, is calculated. That is, we can regard the photometric linearization as a problem to find one correct value calculated by only diffuse reflection factors from among a lot of candidates.

In order to calculate a candidate of the coefficients, three pixels are randomly selected from base images $\boldsymbol{I}_1, \boldsymbol{I}_2, \boldsymbol{I}_3$, and each input image $\boldsymbol{I}_k$. Note that same pixels are selected from every image. A set of coefficients $\hat{\boldsymbol{c}}_k$ is calculated from the intensities of the pixels. By the iteration of this process, a lot of candidate coefficients are obtained.

On the other hand, in order to calculate a candidate of the linearized intensities, three images are randomly selected from the input images. If the coefficients $\boldsymbol{c}_k$ have already been correctly calculated, the intensities $\hat{\boldsymbol{i}}_p^L$ in the linearized base images at pixel p can be easily calculated. By the iteration of this process, a lot of candidate intensities of the linearized base images are obtained.

3.4 Introducing the Criterion for Classification

In order to find a correct value from the numerous candidates calculated by iteration of random sampling, the previous method[12] iterates the estimation of the center of gravity and outlier elimination. However, the algorithm based on a principle of majority has weaknesses. Since the center of gravity may be affected by outliers, an incorrect candidate may be selected because of shadows. So the process tends to be unstable.

Now we propose a new algorithm which can accurately determine the correct value from the numerous candidates. Let's consider the reason why candidates become isolated outliers. That is, we have to check the photometric factors of inliers and outliers. Therefore, we introduce the criterion for classification into the photometric linearization process.

If a candidate is correct, each pixel is classified into the defined factors (D, S, A, and C) by Eq.(2). Any pixel is never classified into the undefined factor (U). Each candidate is evaluated based on the number of pixels which are classified into the defined factors. The candidate which has the maximum number of pixels can be regarded as the correct value.

Basically, the evaluation is based on the defined factors. The specular reflections are, however, excepted from the defined factors. The specular reflection occupies a large area in Fig.(2). If we regard S as the defined factor, incorrect candidates may be accepted. Since the size of the specular region is relatively small in images, we can ignore specular factors in this evaluation. Hence, we evaluate pixels that are classified into diffuse reflection, attached shadow, and cast shadow by

$$Classifiable(k,p) = \begin{cases} 1 & if\ (Region(k,p) = D \cup A \cup C) \\ 0 & if\ (Region(k,p) = S \cup U) \end{cases}. \tag{4}$$

3.5 Evaluation of Candidates

In this section, we present the detailed algorithm to evaluate candidates. For each candidate $\hat{\boldsymbol{c}}_k$ of a set of coefficients, the k-th input image $\boldsymbol{I}_k$ is linearized to $\boldsymbol{I}_k^L$ by the linear combination of the three base images $\boldsymbol{I}_1$, $\boldsymbol{I}_2$, and $\boldsymbol{I}_3$. If $\hat{\boldsymbol{c}}_k$ is correct, Eq.(4) becomes 1 for almost all pixels. Hence, we define the following function to evaluate candidates of the coefficients $\hat{\boldsymbol{c}}_k$.

$$Support^C(k) = \sum_p Classifiable(k,p) \tag{5}$$

On the other hand, the linearized intensities $i^L_{(k,p)}$ are calculated by the linear combination using coefficients $\boldsymbol{c}_k$ for each candidate $\hat{\boldsymbol{i}}^L_p$. If $\hat{\boldsymbol{i}}^L_p$ is correct, Eq.(4) becomes 1 for almost all input images. Hence, we define the following function to evaluate candidates of the linearized intensities.

$$Support^L(p) = \sum_k Classifiable(k,p) \tag{6}$$

The $Support^C(k)$ and $Support^L(p)$ are used to calculate the number of pixels which are classified into valid factors. We can regard the candidates for which the function $Support^C(k)$ or $Support^L(p)$ returns the maximum as the correct value. By using the estimated coefficients $\boldsymbol{c}_k$ and intensities $\boldsymbol{i}_p^L$ in the linearized base images, the accuracy of the photometric linearization can be improved.

3.6 Comparison with the Previous Method [12]

It is noted that the proposed method takes the physical photometric phenomena into account, and considers the photometric factors of outliers, while the previous method [12] is based on only the statistical framework. Therefore, the accuracy can be improved especially in shadow regions.

One may think that if we simply modify [12] so that pixels below the threshold T_s are excluded as outliers, the accuracy can be improved. By ignoring dark regions, similar results may be acquired. However, the new method can analyze the reason of shadows and classify the outliers into two types of shadow.

4 Experimental Results

For the experiments, we used three kinds of materials that have different reflection properties. A ceramic cup (Fig.4) is an example of rough glossy objects, a pot (Fig.7) is an example of very shiny objects, and a marble sphere (Fig.8) is an example of complex reflections such as sub-surface scattering.

4.1 Photometric Classification

We took twenty-four images under various lighting directions in a darkroom keeping a halogen light away from the ceramic cup as shown in Fig.4. Since this cup has a concave surface, some pixels are not illuminated in a number of the input images.

Figure 5 shows three base images selected from input images. (a) shows original base images. (b) and (c) show the results of the photometric linearization. Since the linearized images have negative values, a zero level is expressed as a

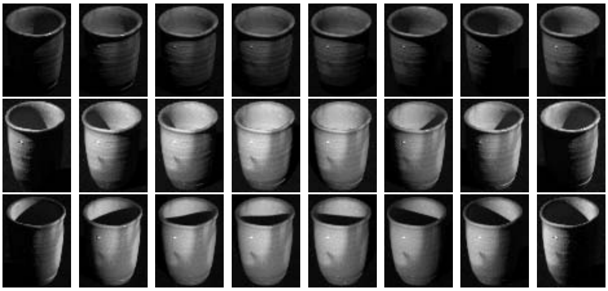
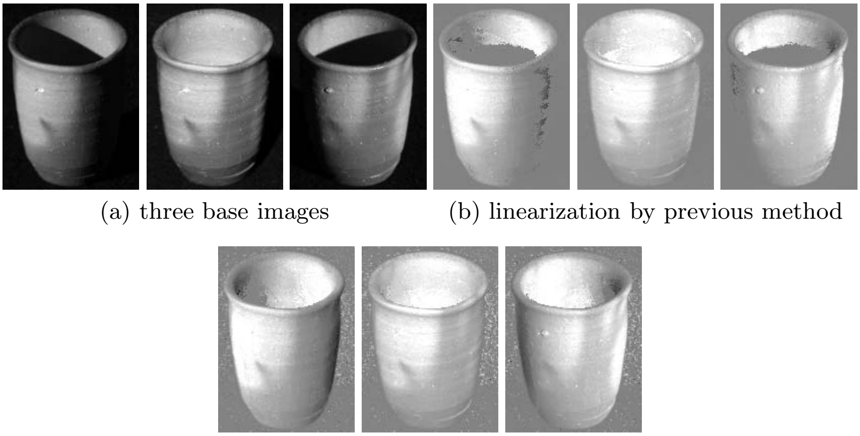


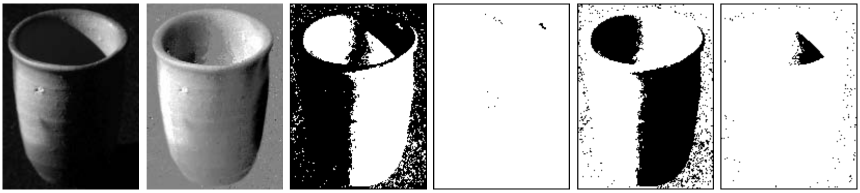
Fig. 4. Input images taken under various lighting directions (cup: twenty-four images)



(a) three base images (b) linearization by previous method

(c) linearization by proposed method

Fig. 5. Linearized base images



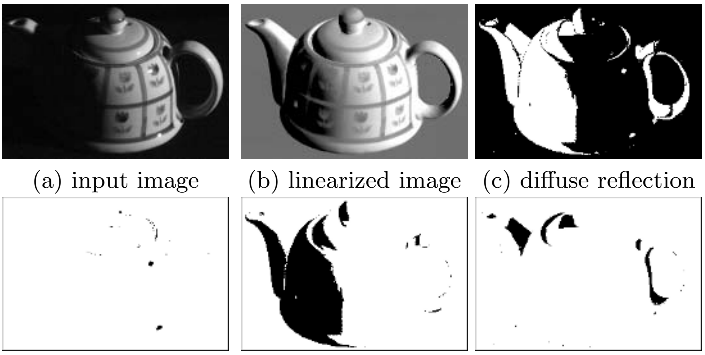
(a) (b) (c) (d) (e) (f)

Fig. 6. Classification results of the photometric factors (cup). (a): an input image, (b): linearized image, (c): diffuse reflections, (d): specular reflections, (e): attached shadows, (f): cast shadows.

gray intensity. (b) shows the results of the previous method. Many pixels are incorrectly linearized to be zero, because the previous method is strongly affected by cast shadows. (c) shows the results of the new method based on the classification criterion. We can see that the base images are correctly linearized even if some pixels are not illuminated in a number of the input images.

Figure 6 shows the results of the photometric classification. (a) is an input image, and (b) is the linearized image. Comparing (a) and (b), each pixel was classified into (c) diffuse reflections, (d) specular reflections, (e) attached shadows, and (f) cast shadows. Although attached shadows and cast shadows cannot be classified by a simple threshold, the proposed method can distinguish them.

Next, we applied our method to a glossy object having complex shape. Figure 7(a) shows an example of twenty-four images. (b) is the result of the photometric linearization. (c), (d), (e), and (f) show the results of classification as diffuse reflections, specular reflections, attached shadows, and cast shadows, respectively. Each pixel can be classified into a suitable photometric factor even if the target object has a complex shape.



(a) input image (b) linearized image (c) diffuse reflection

(d) specular reflection (e) attached shadow (f) cast shadow

Fig. 7. Classification results of a glossy pot

4.2 Photometric Stereo

Next, we show that the photometric linearization can be used for the preprocess of the photometric stereo[1]. We took twenty-four images of a marble sphere under various lighting directions (Fig.8). A part of the surface is not illuminated by obstacles, and complex reflections including subsurface scattering are observed.

After the photometric linearization, the 3-D shape was reconstructed by photometric stereo. Because the lighting directions are unknown, the surface normals cannot be uniquely determined[13]. Therefore, the surface normals are adjusted by the affine transformation to be symmetric around the center of the sphere. Fig.9(a) is a true shape obtained by manual measurement, (b) and (c) are the reconstructed shapes by the previous method and the proposed method, respectively. The previous method failed in the reconstruction due to shadows. On the other hand, new method can correctly linearize and reconstruct at the entire

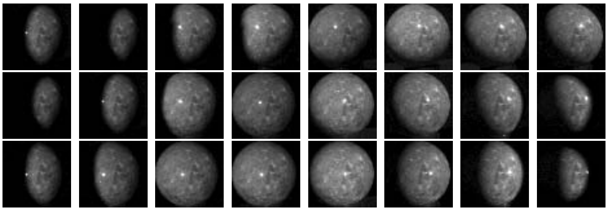
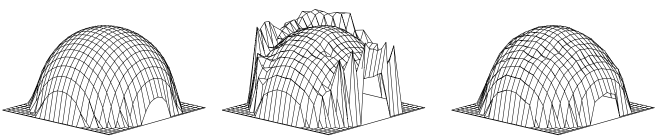


Fig. 8. Input images taken under various lighting directions (sphere)



(a) true shape (b) previous method (c) proposed method

Fig. 9. Reconstructed 3-D shapes

sphere. This result indicates that the photometric linearization method can be applied to objects which have complex BRDFs.

5 Conclusions

In this paper, we proposed a new photometric classification method based on the photometric linearization. While the photometric linearization was originally proposed for generating images under the arbitrary lighting direction, we showed that the method can also be used for the classification of photometric factors. We have improved the accuracy of the photometric linearization method by introducing the classification criterion into the linearization process.

The photometric linearization has an important role as a fundamental technique of computer vision such as photometric stereo and shape-from-shading. We confirmed that our method can be applied for a variety of materials, and that the photometric stereo becomes robust to shadows by applying the photometric classification as a preprocessing. In the future, we intend to analyze more complex factors such as interreflection.

This research was supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (A), 17680018.

References

1. R.J.Woodham: Photometric Stereo, MIT AI Memo, (1978).
2. S.Shafer: Using color to separate reflection components, Color Research and Applications, Vol.10, pp.210-218, (1985).
3. G.Klinker, S.Shafer and T.Kanade: The measurement of highlights in color images, IJCV, Vol.2, No.1, pp.7-32, (1988).
4. Y.Sato and K.Ikeuchi: Temporal-color space analysis of reflection, JOSA A,Vol.11, No.7, pp.2990-3002, (1994).
5. Y.Sato, M.Wheeler and K.Ikeuchi: Object Shape and Reflectance Modeling from Observation, Proc. SIGGRAPH'97, pp.379-387, (1997).
6. L.B.Wolff and E.Boult: Constraining Object Features Using a Polarization Reflectance Model, IEEE Trans. PAMI, Vol.13, No.7, pp.635-657, (1991).
7. S.K. Nayar, X. Fang and T.E. Boult: Removal of specularities using color and polarization, Proc. CVPR'93, pp.583-590, (1993).
8. K.Ikeuchi and K.Sato: Determining Reflectance Properties of an Object Using Range and Brightness Images, IEEE Trans. PAMI, Vol.13, No.11, pp.1139-1153, (1991).
9. A.Shashua: Geometry and Photometry in 3D Visual Recognition, Ph.D thesis, Dept. Brain and Cognitive Science, MIT, (1992).
10. P.N.Belhumeur and D.J.Kriegman: What is the Set of Images of an Object Under All Possible Lighting Conditions?, Proc. CVPR'96, pp.270-277, (1996).
11. A.S.Georghiades, D.J.Kriegman and P.N. Belhumeur: From Few to Many: Illumination Cone Models for Face Recognition Under Variable Lighting and Pose, IEEE Trans. PAMI, Vol.23, No.6, pp.643-660, (2001).
12. Y.Mukaigawa, H.Miyaki, S.Mihashi and T.Shakunaga: Photometric Image-Based Rendering for Image Generation in Arbitrary Illumination, Proc. ICCV2001, pp.652-659, (2001).
13. P.N.Belhumeur, D.J.Kriegman and A.L.Yuille: The bas-relief ambiguity, Proc. CVPR'97, pp.1060-1066, (1997).

Material Classification Using Morphological Pattern Spectrum for Extracting Textural Features from Material Micrographs

D. Ghosh[1] and David C. Tou Wei[2]

[1] Department of Electronics & Communication Engineering,
Indian Institute of Technology Guwahati, India 781039
[2] Department of Electrical & Computer Engineering,
National University of Singapore, Singapore 117576

Abstract. In this paper, we address one very important industrial application of computer vision – automatic classification of materials. In our work, we have considered materials that are mixtures of two or more elements. Such materials are called alloys. It is observed at the microscopic level that an alloy is composed of small randomly distributed crystals of varying shapes and sizes called grains. Also, the color and hence the intensity of the grains vary in alloys. Generally, this shape-size-intensity distribution of the grains is different for different materials. This means micrographs obtained from different materials form texture-like images that differ from one material to another in appearance. Therefore, in principle, any texture analysis method may be used for material classification. In our method, we propose to extract textural features corresponding to grain geometry and intensity and use them for analysis and classification of alloys. These features are extracted via gray-scale morphological operations and are measured in terms of Size-Intensity-Diagram (SID) and Tri-variate Pattern Spectrum (TPS) coefficients. In our experiments, we achieved 83.43% and 89.43% classification accuracies in cases of SID and TPS, respectively. This demonstrates the effectiveness of the proposed method for material classification which in turn confirms that our choice of features is indeed appropriate for the purpose.

1 Introduction

In recent years, Computer Vision has been extensively used in real world systems for commercial, industry and military applications. Some of these applications include industrial automation, biometrics, 3D modelling, video surveillance, classification and recognition, document analysis, medical analysis, human-computer interaction, robotics and so on. In the field of industrial automation, its applications include nondestructive quality and integrity inspection, on-line measurements, etc. thereby aiding the process of manufacturing and inspection. Consequently, computer vision related technologies have started migrating from academic institutions to industrial laboratories.

The objective of this paper is automatic classification of materials which may find application in industry and material science research. However, in this paper,

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 623–632, 2006.

we do not intend to develop any new algorithm for material classification but to build up a system that will view a material sample at microscopic level and will subsequently classify it on the basis of some visual features extracted from the micrograph making use of some existing image processing and computer vision techniques.

In our present work, we have considered materials that are mixtures of two or more elements. Such materials are called *alloys*. Different elements mixed in different proportions give different types of alloys. It is observed at the microscopic level that an alloy is composed of small randomly distributed crystals of varying shapes, sizes and colors called grains. This means a material micrograph obtained from an alloy resembles a texture image in which the grains form the texels (texture elements). It is also observed that the shape-size-color distribution of the grains generally differs from one material to another. As a consequence, texture images obtained from the micrographs of different types of materials generally look different in appearance. Therefore, in principle, any texture analysis method may be used for material classification. Based on this principle, some texture-based material classification schemes had been proposed in [1], [2], [3], [4] and [5]. However, these methods do not take into account the grain geometry and color which otherwise seem to be the most appropriate characterizing features in the context of material classification. On the other hand, the structure of the texture primitive elements (texels) is one very useful and important feature that may be used for the purpose of texture analysis and classification. Therefore, it makes sense to classify materials by extracting textural features corresponding to grain shape and size from the texture-like material micrographs and then apply any available texture classification scheme.

It has been demonstrated through research in material science that the shape and size of the grains composing a material provide important information necessary for characterizing the material, as mentioned in [6] and [7]. In view of this, an earlier attempt to classify materials on the basis of grain size was proposed in [8]. The method involves grain boundary detection and moment calculation. Another efficient tool for shape-size analysis used frequently in image processing and computer vision applications is the mathematical morphology [9]. This is mainly due to its capability in extracting grain geometry and structural information efficiently. Accordingly, some morphological approaches for shape-size based texture analysis were developed in [10], [11], [12] and [13]. Consequently, any of these texture analysis methods may be used for material classification. One such method for material grain size determination using morphological texture analysis is given in [14]. But, all these methods are based on shape-size analysis only and hence are suitable only in cases where color information does not play any significant role.

Apart from grain geometry, another important property that distinguishes one material from another in appearance is the color. An impure material, for example an alloy, when viewed at the microscopic level will show variation in grain color depending on the concentration and nature of different types of crystals composing the material. As a result, a brilliantly white pure material may become cream, grey, pink, brown, or even red due to impurities contained in the crystal structure even in trace amounts. Therefore, extraction of grain color information, in addition to grain shape and size, is equally important for achieving better accuracy in material classification. However, in order to reduce computational complexity, in our work we use monochrome images only where grain color variation manifests as intensity variation in the micrographs. Accordingly, in our work we use gray-scale morphology which is capable of deriving information regarding intensity variation, in addition to shape and size.

2 Proposed Method

Mathematical morphology is an useful tool in many image processing applications that involve shape analysis. In particular, the Pattern Spectrum proposed by Maragos [15] gives us the size distribution of objects within a given image. Extension of the Pattern Spectrum to gray images is the Size Intensity Diagram (SID) [16] which gives a breakdown of the size and gray-level distribution of objects in an image. Another variant of the basic Pattern Spectrum is the Bivariate Pattern Spectrum (BPS) [17] which yields the shape-size distribution in true sense, while the Tri-variate Pattern Spectrum (TPS) [18] is the extension of BPS to gray images. TPS generates the size, gray-level and shape distribution under a single framework. In this paper, we now propose to build up a material classification system based on texture analysis using two variants of the basic Pattern Spectrum, viz., Size-Intensity Diagram and Tri-variate Pattern Spectrum, that give information about the shape, size and intensity variation in a gray image.

2.1 Basic Morphological Operations on Binary Images

The two basic operations in morphology are *dilation* and *erosion*. Given a 2-dimensional image, the object(s) present in it may be represented as a set **A** whose elements are the coordinates of the object pixels. Therefore, **A** is a set in a 2D Euclidean space $\Re^2$, i.e., $\mathbf{A}=\{(a_x, a_y)\}$ where (a_x, a_y) are the coordinates of the object pixels. Let, **B** be another set in $\Re^2$ given as $\mathbf{B}=\{(b_x, b_y)\}$. Then dilation and erosion of **A** w.r.t. **B** are defined as

$$\textit{Dilation:} \quad \mathbf{A} \oplus \mathbf{B} = \bigcup_{(b_x,b_y)\in\mathbf{B}} \left\{(a_x, a_y) + (b_x, b_y) \;\middle|\; (a_x, a_y) \in \mathbf{A}\right\} \ , \tag{1}$$

$$\textit{Erosion:} \quad \mathbf{A} \ominus \mathbf{B} = \bigcap_{(b_x,b_y)\in\mathbf{B}} \left\{(a_x, a_y) - (b_x, b_y) \;\middle|\; (a_x, a_y) \in \mathbf{A}\right\} \ . \tag{2}$$

The set **B** is called the structuring element (SE). Combinations of dilation and erosion give two other morphological operations as follows:

$$\textit{Opening}: \qquad O(\mathbf{A},\, \mathbf{B}) = \mathbf{A} \circ \mathbf{B} = (\mathbf{A} \ominus \mathbf{B}) \oplus \mathbf{B}\ , \tag{3}$$

$$\textit{Closing}: \qquad C(\mathbf{A},\, \mathbf{B}) = \mathbf{A} \bullet \mathbf{B} = (\mathbf{A} \oplus \mathbf{B}) \ominus \mathbf{B}\ . \tag{4}$$

The opening operation acts as a morphological filter in the sense that it retains only those object(s) where the SE can fit in and eliminates the remaining object(s). Closing operation is essentially the opening of the complemented input.

2.2 Pattern Spectrum

A quantitative measure for the size distribution of the objects in an image is the Pattern Spectrum. The number of pixels in the set obtained by subtracting the opened objects from the original one gives the area of those objects that cannot contain the SE. Thus, iterative application of the morphological opening and the measurement of the residues, while increasing the size of the SE, gives the size distribution of the objects contained in the given image. So, if **A** is the set representing the objects in a given 2D image, then following [9] and [19] the pattern spectrum or *pecstrum* may be defined as

$$PS_{n\mathbf{B}}(\mathbf{A}) = \frac{1}{Mes(\mathbf{A})}\Big[Mes(\mathbf{A} \circ n\mathbf{B}) - Mes(\mathbf{A} \circ (n+1)\mathbf{B})\Big]\ , \tag{5}$$

where $Mes(\cdot)$ denotes the finite set cardinality and $n\mathbf{B}$ is the expanded SE of size n (n is any integer in the range 0 to $+\infty$) obtained by dilating **B** iteratively for $(n-1)$ times, i.e.,

$$n\mathbf{B} = \mathbf{B} \underbrace{\oplus \mathbf{B} \oplus \ldots \oplus \mathbf{B}}_{n-1 \quad times}\ . \tag{6}$$

2.3 Bivariate Pattern Spectrum

The pattern spectrum defined above, does not convey the information about the shapes of the objects present in the image. This drawback may be overcome by using Bivariate Pattern Spectrum (BPS). Unlike the usual Pattern Spectrum described above, the size of the SE is increased in vertical and/or horizontal direction so as to vary both the size and the shape of the SE. Thus, the residues so obtained at all stages of opening and subsequent subtraction give the shape distribution of the objects to some extent, in addition to the size description. Therefore, BPS is the generalization of the usual Pattern Spectrum and is the true shape-size descriptor for the objects present in the given binary image. Accordingly, the BPS is defined as

$$\begin{aligned} &BPS_{((n_x,n_y)\mathbf{B})}(\mathbf{A}) \\ &\quad = \tfrac{1}{Mes(\mathbf{A})}\left\{Mes(\mathbf{A} \circ (n_x,n_y)\mathbf{B}) + Mes(\mathbf{A} \circ (n_x+1,n_y+1)\mathbf{B})\right. \\ &\qquad \left. - Mes(\mathbf{A} \circ (n_x+1,n_y)\mathbf{B}) - Mes(\mathbf{A} \circ (n_x,n_y+1)\mathbf{B})\right\}\ , \end{aligned} \tag{7}$$

where $(n_x, n_y)\mathbf{B}$ is the SE of dimension n_x by n_y.

2.4 Basic Morphological Operations on Gray Images

A gray scale image is defined as a 2D function $f(a_x, a_y)$ where (a_x, a_y) is the coordinate of a pixel in the image and $f(a_x, a_y)$ gives the corresponding pixel intensity. The object present in the image, hence, may be defined in the form of a set of triples $\mathbf{A} = \{(a_x, a_y, a_g)\}$ where (a_x, a_y) are the object pixels and $a_g = f(a_x, a_y)$. The gray scale structuring element $\mathbf{B}$ may also be defined in a similar way in the form of a set $\{(b_x, b_y, b_g)\}$. The morphological operations on the image $\mathbf{A}$, hence, are defined in [19] and [20] as

Gray scale dilation:

$$\mathbf{A} \oplus \mathbf{B} =_{(b_x,b_y,b_g)\in\mathbf{B}}^{\text{EXTSUP}} \left| \{(a_x, a_y, a_g) + (b_x, b_y, b_g) \right| (a_x, a_y, a_g) \in \mathbf{A}\Big\} \ , \qquad (8)$$

Gray scale erosion:

$$\mathbf{A} \ominus \mathbf{B} =_{(b_x,b_y,b_g)\in\mathbf{B}}^{\text{INF}} \left| \{(a_x, a_y, a_g) - (b_x, b_y, b_g) \right| (a_x, a_y, a_g) \in \mathbf{A}\Big\} \ . \qquad (9)$$

The opening and closing operations are defined as their counter parts in binary operations.

2.5 Size Intensity Distribution

Using the idea of the Pattern Spectrum, and incorporating gray level (intensity) information, Size-Intensity Diagram (SID) is obtained as

$$SID_{((n,g)\mathbf{B})}(\mathbf{A}) = \frac{1}{Mes(\mathbf{A})} \{Mes(\mathbf{A} \circ (n,g)\mathbf{B}) + Mes(\mathbf{A} \circ (n+1, g+1)\mathbf{B}) \\ -Mes(\mathbf{A} \circ (n+1,g)\mathbf{B}) - Mes(\mathbf{A} \circ (n, g+1)\mathbf{B})\} \ , \qquad (10)$$

where $(n, g)\mathbf{B}$ is a flat SE of size n with gray level g.

2.6 Tri-variate Pattern Spectrum

Using the above relations for the gray scale morphological operations, the idea of BPS is extended to Tri-variate Pattern Spectrum (TPS) so as to obtain the shape-size description in a gray scale image. In the TPS, the shape of the structuring element $\mathbf{B}$ is varied via separate expansion in the x and y dimensions together with the variation of gray levels of the structuring element. The TPS defined at each gray level g is defined as

$$TPS_{((n_x,n_y,g)\mathbf{B})}(\mathbf{A}) \\ = \tfrac{1}{Mes(\mathbf{A})} \{Mes(\mathbf{A} \circ (n_x, n_y, g)\mathbf{B}) + Mes(\mathbf{A} \circ (n_x + 1, n_y + 1, g)\mathbf{B}) \qquad (11) \\ -Mes(\mathbf{A} \circ (n_x + 1, n_y, g)\mathbf{B}) - Mes(\mathbf{A} \circ (n_x, n_y + 1, g)\mathbf{B})\} \ ,$$

where $(n_x, n_y, g)\mathbf{B}$ is a flat structuring element of dimension n_x by n_y with gray level g, $g = 1, 2, \ldots, L-1$, L is the number of gray-levels in the image. Gray level $g = 0$ generally corresponds to the inter-grain gaps and cavities and hence is not considered in evaluating the TPS coefficients.

2.7 Material Classification Using SID and TPS

Using Scanning Electron Microscope, microscopic images of materials known as micrographs are obtained. These micrographs are subsequently converted to gray images. As mentioned before, at the microscopic level, it is observed that materials are made up of grain patterns that give texture-like appearance to the micrographs. Also, the shape, size and intensity distribution of grains in one material is generally different from that of another material. This aspect of the micrographs is utilized for the purpose of material classification. In other words, a material may be recognized on the basis of the shape, size and intensity distribution of the grains that the material is composed of. And for the purpose of feature extraction from different materials the SID and TPS seem to be suitable in the present context while classification may be accomplished by employing any gray texture analysis scheme.

As with binary textures, gray-scale morphological approach seems to be an efficient tool in gray texture analysis involving grain shape analysis. One such morphological approach to gray texture analysis is given in [21] in which a model of the elementary particles that form a texture is obtained by applying pattern spectrum with gray-scale structuring elements. However, in this method, the extra step necessary to determine optimal structuring elements increases the computational overhead. In later times, a TPS-based texture analysis scheme had been developed in [22] which may be applied on material micrographs so as to accomplish material classification. However, TPS is generally computationally expensive. A relatively less complex scheme may be to use SID in place of TPS but at the cost of classification accuracy. The set of SID or TPS coefficients forms the set of textural features corresponding to shape, size and intensity of the material grains and is subsequently used in the classification stage.

3 Experimental Results

In our experiments, we have evaluated the accuracy in classifying different materials by applying texture analysis on material micrographs in which the textural features are measured in terms of SID and TPS coefficients, as proposed in this paper. Seven different types of materials with 250 training and 50 test micrographs per material type are taken. The colored micrographs are converted to gray images with 256 gray levels. The basic structuring element taken is a 3×3 square and a k-NN classifier is used for classification. The different types of materials taken are (A) Copper-Zinc alloy, (B) Steel with 0.1% Carbon, (C) Steel with 0.5% Carbon, (D) Silicon-Carbide (E) Steel with 0.4% Carbon, (F) Steel with 1.25% Carbon, and (G) Ferrite XIV. Figure 1 shows the micrographs for each of these materials, one sample per material type. The classification results obtained in our experiments are given in Table 1 and Table 2.

We see that the proposed material classification scheme using SID and TPS coefficients works well yielding accuracy rate as high as 100% for some materials while the overall recognition rates are 83.43% and 89.43% in cases of SID and TPS, respectively. From Fig. 1, we see that the microscopic views of some materials are so similar (e.g., CuZn and Steel with 0.1% Carbon) that manual discrimination is almost impossible. Even then, our classifier is capable of discriminating them to some extent. We also observe that TPS yields better recognition rate compared to SID, but at the cost of increased computational load. This is because TPS has better shape analyzing capacity than SID.

Table 1. Recognition result in material classification using SID coefficients. Seven different types of materials are taken and our proposed classification method is tested on 50 samples per material type.

Class labels of input test samples	Number of test samples classified to to each of the seven material classes							Recognition Rate in percentage
	A	**B**	**C**	**D**	**E**	**F**	**G**	
A	31	15	0	2	1	1	0	62.0
B	1	45	2	1	0	1	0	90.0
C	1	1	36	6	0	6	0	72.0
D	0	0	0	50	0	0	0	100.0
E	0	0	0	0	50	0	0	100.0
F	0	0	0	4	16	30	0	60.0
G	0	0	0	0	0	0	50	100.0
	Average Recognition Rate							83.43

Table 2. Recognition result in material classification using TPS coefficients. Seven different types of materials are taken and our proposed classification method is tested on 50 samples per material type.

Class labels of input test samples	Number of test samples classified to to each of the seven material classes							Recognition Rate in percentage
	A	**B**	**C**	**D**	**E**	**F**	**G**	
A	35	8	1	2	0	4	0	70.0
B	0	40	1	1	1	7	0	80.0
C	0	0	44	0	0	6	0	88.0
D	0	0	0	50	0	0	0	100.0
E	0	0	0	0	50	0	0	100.0
F	0	0	0	5	0	45	0	90.0
G	0	0	0	0	0	1	49	98.0
	Average Recognition Rate							89.43

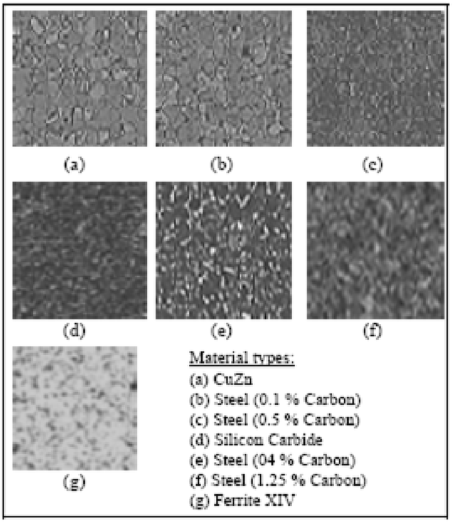


Fig. 1. Micrographs of the seven different types of materials used in our experiment. For computational simplicity the actual color micrographs have been converted to gray images as shown here. The texture like appearance of the micrographs can be observed in the figures.

4 Conclusion

In this paper, we have explored the potentiality of using morphological pattern spectrum for material classification. Two variants of the morphological Pattern Spectrum, namely the Size-Intensity-Diagram (SID) and the Tri-variate Pattern Spectrum (TPS), are used for extracting textural features from the texture-like microscopic images of the materials and are then used for classification in a manner similar to any texture analysis and classification method. Based on our experimental results, it is found that the SID and TPS coefficients, in particular the TPS coefficients, are indeed good measure for the textural features corresponding to the shape-size-intensity distribution of the material grains in the micrographs. Hence our proposed method may be reliably used for material analysis, process control, etc.

The scheme described in this paper may be extended to some applications as follow.

1. *Material inspection*: The proposed method may be used for locating any defect, fault, presence of impurities, etc. in a material sample. The shape-size-intensity distribution of the material grains may be extracted by scanning the input sample thoroughly. Deviation from this distribution measure at any point in the sample will indicate defect or presence of impurity at that location.

2. *Material characterization*: The structure, size and color of the grains determine important physical properties of a material. For example, high aspect-ratio in grain size indicates good mechanical reinforcing effect. Materials composed of coarse sized grains generally detract from mechanical reinforcement, segregate and settle quickly, affect the processing and quality of end-use products, lead to higher abrasion, and affect surface finish. On the other hand, excessive amounts of fine grains can lead to ineffective mechanical reinforcement, high resin consumption as a filler, and problems with materials handling. Also, the density of a material may be assessed by evaluating the number of grain pixels in a micrograph. Similarly, distribution of grain intensity (or color) may be used to assess the concentration of different elements in an alloy.

References

1. Dana, K., van Ginneken, B., Nayar, S., Koenderink, J.: Reflectance and texture of real world surfaces. ACM Trans. Graphics **18** (1999) 1–34
2. Chantler, M., McGunnigle, G., Wu, J.: Surface rotation invariant texture classification using photometric stereo and surface magnitude spectra. In: Proc. 11th British Machine Vision Conf., Bristol, UK (2000) 486–495
3. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three dimensional textons. Intl. J. Computer Vision **43** (2001) 29–44
4. Varma, M., Zisserman, A.: Classifying images of materials: Achieving viewpoint and illumination independence. In: Proc. 7th European Conf. Computer Vision. Volume 3., Denmark (2002) 255–271
5. Varma, M., Zisserman, A.: Statistical approaches to material classification. In: Proc. 3rd Indian Conf. Computer Vision, Graphics and Image Processing, Ahmedabad, India (2002) 167–172
6. Meloy, T.: Shape characterization of particles – problems and progress. In: Advanced Materials – Application of Mineral and Metallurgical Processing Principles, Society of Mining Engineers of AIME, USA (1990) 195–203
7. Nicoletti, D., Bilgutay, N., Onaral, B.: Power-law relationships between the dependence of ultrasonic attenuation on wavelength and the grain size distribution. J. Acoustical Society of America **91** (1992) 3278–3284
8. Wang, W., Bergholm, F.: On moment-based edge density for automatic size inspection. In: Proc. 9th Scandinavian Conf. Image Analysis. Volume 2., Sweden (1995) 895–904
9. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London (1982)
10. Huet, F., Mattioli, J.: A textural analysis by mathematical morphology transformations: Structural opening and top-hat. In: Proc. Intl. Conf. Image Processing. Volume 3., Switzerland (1996) 49–52
11. Li, W., Hease-Coat, V., Ronsin, J.: Robust morphological features for texture classification. In: Proc. Intl. Conf. Image Processing. Volume 3., Switzerland (1996) 173–176
12. Asano, A.: Texture analysis using morphological pattern spectrum and optimization of structuring element. In: Proc. 10th Intl. Conf. Image Analysis and Processing, Italy (1999) 209–214

13. Singh, S., Kumar, V., Ghosh, D.: Binary texture analysis and classification using bi-variate morphological pattern spectrum. In: Proc. Intl. Conf. Imaging Science, Systems and Technology. Volume 2., Las Vegas (2001) 790–795
14. Rautio, H., Silven, O.: Average grain size determination using mathematical morphology and texture analysis. In: Proc. 6th IAPR Workshop on Machine Vision Applications, Japan (1998) 506–509
15. Maragos, P.: Pattern spectrum and multiscale shape representation. IEEE Trans. Pattern Analysis & Machine Intelligence **11** (1989) 701–715
16. Trettel, E., Lotufo, R.: The size-intensity diagram: A gray-scale granulometric analysis tool. In: Proc. 9th Brazilian Symp. Computer Graphics and Image Processing, Caxambu, Brazil (1996) 259–264
17. Ghosh, P., Chanda, B.: Bi-variate pattern spectrum. In: Proc. 11th Brazilian Symp. Computer Graphics and Image Processing, Rio de Janeiro, Brazil (1998) 476–483
18. Athreya, G., Ghosh, D.: Trivariate pattern spectrum: A shape-size descriptor for gray scale images. In: Proc. 8th Natl. Conf. Communications, Mumbai, India (2002) 40–44
19. Giardina, C., Dougherty, E.: Morphological Methods in Image and Signal Processing. Prentice-Hall, Englewood Cliffs, New Jersey (1998)
20. Sternberg, S.: Grayscale morphology. Computer Vision, Graphics and Image Processing **35** (1986) 333–355
21. Asano, A., Miyagawa, M., Fujio, M.: Texture modelling by optimal gray scale structuring element using morphological pattern spectrum. In: Proc. 15th Intl. Conf. Pattern Recognition. Volume 3., Spain (2000) 475–478
22. Athreya, G., Ghosh, D., Chakrabarti, I.: Texture classification using morphological trivariate pattern spectrum for texture description. In: Proc. Intl. Conf. Imaging Science, Systems and Technology. Volume 2., Las Vegas (2002) 830–836

A Hierarchical Framework for Generic Sports Video Classification

Maheshkumar H. Kolekar and Somnath Sengupta

Electronics and Electrical Communication Engineering Department,
Indian Institute of Technology,
Kharagpur-721302, West Bengal, India
{mhkolekar, ssg}@ece.iitkgp.ernet.in

Abstract. A five layered, event driven hierarchical framework for generic sports video classification has been proposed in this paper. The top layer classifications are based on a few popular audio and video content analysis techniques like short-time energy and Zero Crossing Rate (ZCR) for audio and Hidden Markov Model (HMM) based techniques for video, using color and motion as features. The lower layer classifications are done by applying game specific rules to recognize major events of the game. The proposed framework has been successfully tested with cricket and football video sequences. The event-related classifications bring us a step closer to the ultimate goal of semantic classifications that would be ideally required for sports highlight generation.

1 Introduction

Event-based storage and retrieval of sports video sequences and automated generation of highlights are highly demanding topics, because of their popularity and commercial importance. Therefore, there has been a widespread studies in the field of sports video classification. E. Kijak et. al. [1] have presented the use of HMM for the structure analysis of Tennis video. J. Assfalg et. al. [2] have worked upon football video classification using camera motion and player's location. L. Duan et. al. [3] have proposed the color characterization model for sports video indexing and browsing. L. Xie et. al. [4] have proposed an algorithm for parsing the structure of soccer sports video. These works provide fairly compressive, solutions to the task outlined, however the challenge of developing a solution or scheme that can reveal common structures of multiple events across multiple domains remains under-investigated. In practice though, such a scheme could not exist without some limit of domain constraint, i. e. the design of common feature extraction metrics applied to two vastly different sports types. On the other hand it is important to avoid becoming too context specific. With this trade-off in mind, our research is aimed towards designing techniques such that they can be globally applied to all sports types, which come under the umbrella of 'ball and field sports'. Recently, unified general frameworks were proposed in [5],[6]. In their work, excitements were extracted for highlight generation, but

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 633–642, 2006.

detailed event classification was not carried out. The sports video classification schemes proposed till date fail to respond to action-based queries, such as "extract the goal clips out of this football sequence", or "find out when the batsman got run-out in this cricket video", etc. Such queries may be always needed for editing and retrieval.

Very recently, W. Hsu et. al. [7] have presented the scheme for the fusion of multimode features for TRECVID news video classification. L. Chaisorn et. al. [8] has proposed two-level framework for news classification. In these approaches, they segment the full video into video shots and then clustering the shots to generate the concept hierarchies. In contrast to these method, we have used the top down approach, which permits us to avoid clustering and consequently improves the classification accuracy and also maintains the temporal order of shots.

Successful solution of this problem has to address two basic issues; (a) the classification should be event-related, and (b) such events should fit well within a generic framework that can be used for any popular sports. The first issue is not easy to solve since the classification schemes do not use semantic information directly, but use basic clues like color, motion etc. At the top layers, we use these clues to achieve basic classifications and for the subsequent layers, we apply event-driven rules from the top layers to recognize the events. We successfully tried our approach with two popular games - football and cricket, but it is applicable to other games as well. To address the second issue, we have proposed a generic event representation framework, that is simple, hierarchical in nature and makes indexing and retrieval process easy, and straightforward.

The fundamental problem associated with the top-layer video classification is the large volume of data that we have to deal with. In every game, there are moments of excitement, with relatively dull periods in between. Only moments of excitements qualify for inclusion in the highlights and the dull periods, which are often lengthy in terms of number of frames, need to be filtered out. Excitements are always accompanied by significant audio content resulting from spectators' cheering, increase in the audio level of the commentators' voice etc. After carrying out large set of experiments, we conclude that audio serves as the most basic clue to filter out the dull contents and extract clips that may qualify for inclusion in highlights. We have used two popular audio content analysis techniques- short-time energy and ZCR for extracting possible moments of excitements. However, all excitements detected through audio features may not correspond to game excitements. Even commercial clips are sometimes associated with excitement type of audio contents and these must be detected and filtered out. In the next layers, video features, like color and motion have been used for classification. One of the major characteristics of sports video is that the sequences are highly structured in the sense that the number of events is usually limited in number and there are repetitive transitions, often back and forth between those events. Using a large number of video sequences for training, we have derived the scene structure in terms of the transition probabilities from one event to the other and trained a HMM model [9] for classification of the events.

2 Hierarchical Classifications

The tree diagram shown in Fig 1 is our proposed generic framework for the sports video classification. At the top layer, the requirement is to skim the video sequence to a significant extent and extract the possible moments of excitements. As explained in the previous section, audio features serve as a very important clue for this top-layer classification, which is essentially binary - excitement (L1: class-0) and non-excitement (L1: class-1). Of the clips labeled as "excitement" in level-1, some frames show direct game actions (L2: class-0), some display the spectators (L2: class-1), especially after the major events like goal in football and "out" or "sixer" in cricket. The first level audio classification even picks up the commercials (L2: class-2), since often those are presented with exciting tones and background effects added. At the next level, i.e., level-3, the game actions are further sub-classified into real-time events, post-event activities and replay. Every major event is followed by some post-event activities like players' celebrations and finally, replays are presented, while the audio excitements still continue. At level-4, real-time shots (L3: class-0) are classified into actions, based on a set of rules applied on real-time shots and post-event activities. At the next level, the actions are further classified into a set of rule-based sub-actions. The definition of action and sub-action can be specialized to a specific sports based on specific domain knowledge. For example, an action can be wicket and sub-action can be the type of wicket e.g. bowled, catch etc. in the cricket. In football, goal is an action and the type of goal, e.g. goal by penalty kick, goal by head etc. are sub-actions. The rules applied for action and sub-action detection are game-

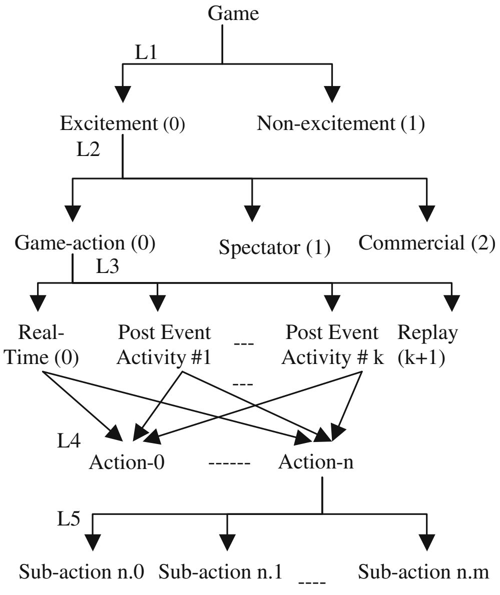


Fig. 1. Tree Diagram of Hierarchical Structure

specific. For example, in cricket, if the real-time actions are followed by fielders' celebrations and batsman's departure in close-up, it is a wicket, otherwise, it is a hit. In football, if the real-time actions are followed by players' celebrations and close-up, it is a goal, otherwise, a goal-miss.

3 Event Detection and Classification

3.1 Excitement Detection at Level 1

We have observed that whenever there is an important activity in the game, there is a corresponding increase in audio energy. We have used two popular audio content analysis techniques- short-time energy and ZCR [10] for extracting commercials. A particular video frame is considered as an excitement frame if its audio excitement or ZCR exceeds the threshold. The short time audio energy $E(n)$ and ZCR $Z(n)$ for frame n is computed as follows:

Short-time audio energy

$$E(n) = \frac{1}{V} \sum_{m=0}^{V-1} [x(m)w(n-m)]^2$$

where,

$$w(m) = \begin{cases} 1 \text{ if } \quad 0 \leq m \leq V-1 \\ 0 \text{ otherwise} \end{cases}$$

$x(m)$ is the discrete time audio signal, V is the number of audio samples corresponding to one video frame.

Short-time average zero-crossing rate
In discrete-time signals, a zero crossing is said to occur if successive samples have different signs. The short-time average zero-crossing rate $Z(n)$, as defined below, gives rough estimates of spectral properties of audio signals.

$$Z(n) = \frac{1}{2} \sum_{m=0}^{V-1} |sgn[x(m)] - sgn[x(m-1)]| w(n-m)$$

where,

$$sgn[x(m)] = \begin{cases} 1 \; x(m) \geq 0 \\ -1 \; x(m) < 0 \end{cases}$$

where, and $w(m)$ is a rectangular window. It is observed that audience cheering generally leads to high ZCR.

The strategy for the excitement detection is already explained in our previous work [10].

3.2 HMM Based Video Classification for Level 2 to 5

We have used HMM model to classify the trimmed video into one of the predefined classes. The class transition diagram is generated by training HMM through a number of sports video sequences and once trained, video shots can be classified into available classes by matching to the models of these classes. Our approach can be summarized as follows:

Step-1: Likelihood computations
Compute the likelihood $l_t(k)$ that the frame-t belongs to the class-k, based on the similarity of the features (such as color, motion etc.) of frame-t with those of class-k.

Step-2: Accumulated Likelihood Computation
Corresponding to the starting frame, the accumulated likelihood $L_t(k)$ for every class and the backtracking indices $A_t(k)$ for every class are initialized as follows:

$$L_1(k) = \alpha \ l_1(k)$$

and

$$A_1(k) = 0 \quad for \ k = 0, 1, 2, 3, ...N$$

where α is a multiplication constant.

For all subsequent frames, the accumulated likelihood and backtracking indices of every class is computed through a dynamic programming based optimum path search and is given by:

$$L_t(k) = \max_{1 \leq i \leq N} (L_{t-1}(i) + c(i,k)) + \alpha \ l_t(k)$$

and

$$A_t(k) = arg \max_{1 \leq i \leq N} (L_{t-1}(i) + c(i,k))$$

In the above equations, $c(i,k)$ indicates the transitional probability from class-i to class-k, determined through training. It is obvious from accumulated likelihood equation that higher value of multiplication factor α contributes to predominance of current likelihood over the accumulated ones.

Step-3: Frame-by-frame classification
Following step-2, the frames are classified individually, starting with the class C_t^* for the last frame of the sequence and continuing through a process of backtracking, as given below

$$C_T^* = arg \max_{1 \leq i \leq N} (L_T(i))$$

and

$$C_t^* = A_{t+1}(C_{t+1}^*),$$
$$where, \ t = T-1, T-2, ..., 1$$

3.3 Rule Based Activity Detection

To bridge the semantic gap, we have used the rule-based approach for level-4 for extracting semantic video concepts. The generic rule can be formed as follows

If{event A is followed by
post event activity ♯ 1 and/or post event activity ♯ 2
and/or· · · · · post event activity ♯ k}
Then {event A belongs to class i}
Else {event A belongs to class j }
Typical Examples of the rules for cricket video:
If {real time (L3: class-0) is followed by:
{Fielders' celebration (L3: class-1)} and/or {Batsman's departure (L3: class-2)}}
Then {action is wicket (L4: class-0)}
Else {action is hit (L4: class-1)}

Such many rules can be generated at different levels of hierarchical structure to extract the semantic concepts of the sports video.

4 Implementation and Results

We have tested our proposed approach using live recording of cricket and football video sequences. We sampled audio at a rate of 44.1 KHz. The performance of excitement detection was tested using the measure detection accuracy η_D, which is the ratio of number of excitement frames correctly detected, to the total number of actual excitement frames. Table 1 and Table 2 presents the detection accuracy for cricket video clip of 5:10 minutes and football video clip of 2:24 minutes respectively. For cricket video clip, we have extracted total 3109 video frames at the rate of 10 frames/second and for football video clip, we have extracted total 725 video frames at the rate of 5 video frames/second to increase computational speed. Fig 2 and Fig 3 show the graphs of audio energy and ZCR vs video frame number for cricket and football video respectively. We observed the average detection accuracy as 98.23% for cricket test video and 100% for football test video.

The overall performance of event classification is tested using the measure classification efficiency η_c, which is the ratio of the number of frames correctly classified to the total number of frames belonging to that particular class. Table 3

Table 1. Cricket video classification at Level 1 for various values of window size

Window size (sec)	Actual ♯ of excitement frames	♯ of excitement frames correctly detected	η_D %
5	1137	1094	96.22
10	1137	1110	97.63
20	1137	1132	99.56
40	1137	1129	99.30
50	1137	1119	98.42

Table 2. Football video classification at level 1 for window size of 10 seconds

Activity observed	Actual ♯ of excitement frames	♯ of excitement frames correctly detected	η_D %
Foul	69 (1-68)	69	100
Goal miss	154 (175-329)	154	100
Free kick	140 (535-675)	140	100

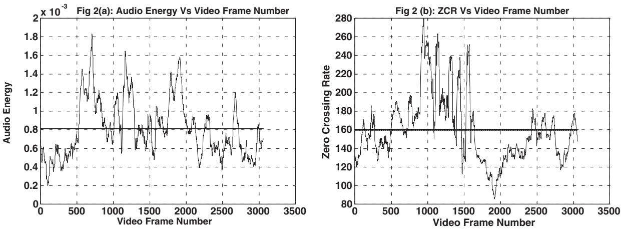


Fig. 2. (a) Audio Energy (b) ZCR Vs Video Frame Number for cricket video sequence

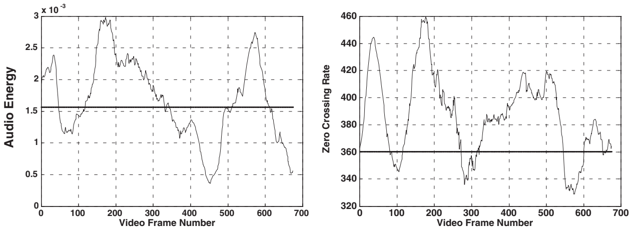


Fig. 3. (a) Audio Energy (b) ZCR Vs Video Frame Number for football video sequence

Table 3. Class Definitions of level-3

Class Number	Cricket	Football
0	Real time	Real time
1	Fielders' Celebration	Players' Celebration
2	Batsman's Departure	Players' Close-up
3	Replay	Replay

indicates our class definitions for level-3 for cricket and football video sequences. Table 4 and 5 represent the classification accuracy of cricket and football videos respectively.

Fig 4 shows boundary frames of the scenes of level-2 of cricket video, where we have used color as a likelihood function, since the color of ground is green and can be easily distinguished from spectator and commercial class. Fig 5 shows boundary frames of the scenes of level-3 of cricket video, where we have used color and motion as a likelihood function, since the color of ground is green and can be easily distinguished from fielders' celebration (where the dominance of

Table 4. Cricket video classification

Level	Beginning-end frame/ total frames/actual class	♯ of frames in observed class 0/1/2/3/..	η_c %	$\bar{\eta}_c$ %
2	525-894/370/0	314/38/18	84.86	94.28
	895-964/70/1	0/70/0	100	
	965-1661/697/2	0/14/683	97.99	
3	525-627/103/0	73/12/18/4	70.87	79.02
	628-733/106/1	2/83/21/8	78.30	
	734-790/57/2	1/1/54/1	94.74	
	791-894/104/3	12/8/11/74	71.15	
4	525-627/103/0	103/0	100	100
5	525-627/103/0.4	4/3/4/4/88	85.44	85.44

Table 5. Football video classification

Level	Beginning-end frame/ total frames/actual class	♯ of frames in observed class 0/1/2/3/..	η_c %	$\bar{\eta}_c$ %
2	175-329/154/0	154/0/0	100	100
	-/0/1	0/0/0	100	
	-/0/2	0/0/0	100	
3	175-198/24/0	19/2/2/1	79.16	87.30
	-/0/1	0/0/0/0	100	
	199-219/21/2	2/2/17/1	80.95	
	220-329/110/3	2/6/4/98	89.09	
4	175-198/24/1	0/24	100	100
5	175-198/24/1.2	3/2/17/2	70.80	70.80

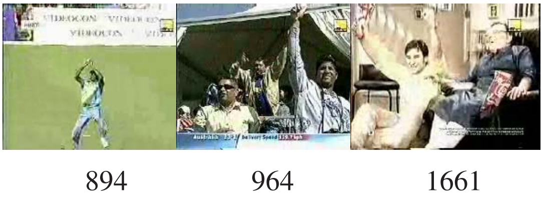


Fig. 4. Boundary frames of the scenes classified into class-0, class-1, and class-2 in level-2 of the cricket video

blue color is observed because our test video contains Indian fielders whose dress color is blue.) and batsman's departure (where the dominance of yellow color is observed because the color of Australian batsman's dress is yellow). We have also used motion as a likelihood to separate the real time action on the ground from the replays. Since the real time action is followed by fielders' gathering, our rule based classifier has declared the event in the cricket test video as a wicket.

Fig 6 shows boundary frames of the scenes of level-2, where we have used color as a likelihood function, since the color of ground is green and can be easily dis-

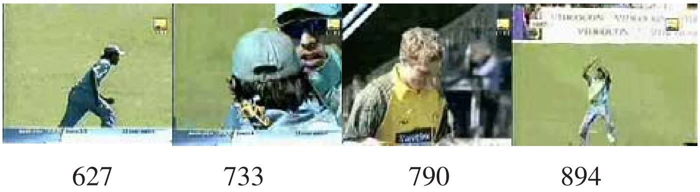


Fig. 5. Boundary frames of the scenes classified into class-0, class-1, class-2, and class-3 in level-3 of cricket video

Fig. 6. Boundary frames of the scenes classified into class-0, class-1 (no frame) and class-2 (no frame) in level 2 of football video

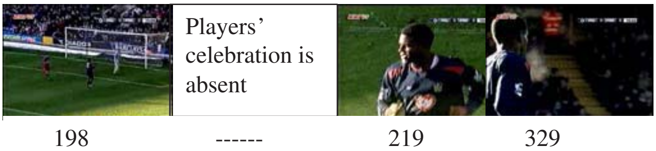


Fig. 7. Boundary frames of the scenes classified into class-0, class-1 (no frame), class-2 and class-3 in level 3 of football video

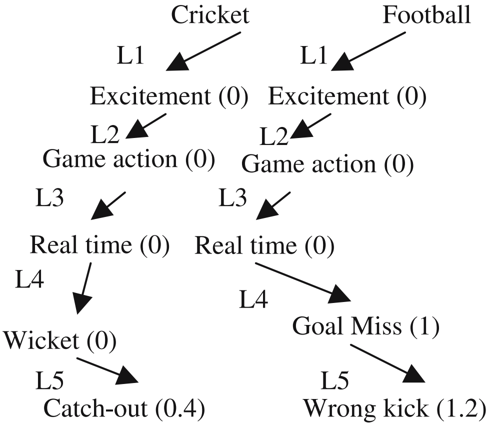


Fig. 8. Tree path followed by (a) cricket, and (b) football test video sequences

tinguished from spectator and commercial class. Fig 7 shows boundary frames of the scenes for level-3 for football video sequence where we have observed that the frames of class-1 are absent. This indicates that the players' celebration is absent. Hence our rule-based classifier has declared this activity as goal miss. Fig 8 shows the tree path followed by cricket and football test video sequences.

5 Conclusion and Future Work

In this paper, we have presented a generic hierarchical framework for sports video classification and successfully applied it to cricket and football. Integrating audio and video features for classifier not only reduces the cost of processing data drastically, but also increases the classifier accuracy significantly. The proposed modeling is readily applicable to media database management applications, where common operations such as indexing, retrieval, logging, annotation and highlights, etc can all benefit from the breakdown of a video into the smaller segments.

References

1. Kijak, E., Gravier, G., Gros, P., Oisel, L., Bimbot, F.: Hmm based structuring of tennis videos using visual and audio cues. in Proc. of Int. Conf. on Multimedia and Expo **3** (2003) 309–312
2. Assfalg, J., Bertini, M., Bimbo, A.D., Nunziati, W., Pala, P.: Detection and recognition of football highlights using hmm. in 9th Int. Conf. on Electronics, Circuits and Systems **3** (2002) 1059–1062
3. Duan, L., Xu, M., Tian, Q., Xu, C.: Nonparametric color characterisation using mean shift. Proc. of 11^{th} ACM Int. Conf. on Multimedia (2003) 243–246
4. Xie, L., Chang, S.F., Divakaran, A., Sun, H.: Structure analysis of soccer video with hidden markov models. in Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing (2002)
5. Baoxin, L., Pan, H., Sezan, I.: A general framework for sports video summarization with its application to soccer. in Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing **3** (2003) 169–172
6. Hanjalic, A.: Generic approach to highlights extraction from a sports video. in Proc. of IEEE Int. Conf. on Image Processing **1** (2003) 1–4
7. Hsu, W., Kennedy, L., Huang, C.W., Chang, S.F., Lin, C.Y., Iyengar, G.: News video story segmentation using fusion of multi-level multi-modal features in trecvid 2003. in Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing **3** (2004) 645–648
8. Chaisorn, L., Chua, T.S., Lee, C.H.: The segmentation of news video into story units. in Proc. of Int. Conf. on Multimedia and Expo **1** (2002) 73–76
9. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. in Proc. of IEEE **77** (1989)
10. M.H.Kolekar, Sengupta, S.: Hierarchical structure for audio-video based semantic classification of sports video sequences. in Proc of SPIE Int. Conf. on Visual Communications and Image Processing, Beijing, China **5960** (2005) 401–409

Feature Detection with an Improved Anisotropic Filter

Mohamed Gobara and David Suter

Department of Electrical and Computer Systems Engineering,
Monash University, Clayton, 3800 Victoria, Australia
{osman.gobara, d.suter}@eng.monash.edu.au

Abstract. The problem of detecting local image features that are invariant to scale, orientation, illumination and viewpoint changes is a critical issue in many computer vision applications. The challenges involve localizing the image features accurately in the spatial and frequency domains and describing them with a stable analytical representation. In this paper we address these two issues by proposing a new non-linear scale-space implementation that improves the localization accuracy of the SIFT [3] local features. Furthermore we propose a simple adjustment to the standard SIFT descriptor and show that the modified version is more robust to affine changes.

1 Introduction

Interest point detection is a key issue in many computer vision applications including motion tracking, object recognition and 3D reconstruction. An interest point is any point in the image that is characterized by distinctive neighboring features. This includes L-corners, T-junctions, Y-junctions and highly textured areas. The detection of interest points is a dual stage process, (a) localization and (b) representation. In the localization phase we detect the position and the scale of each interest point and in the representation phase we use an analytical model to describe the local shape or pattern at each interest point. The goodness of a model (i.e. also known as a local descriptor) is measured in terms of its degree of invariance over transformations caused by viewpoint and illumination changes. A good model (i.e. highly invariant descriptor) would identify a local pattern, before and after being transformed, with the same numeric measure.

Schmid and Mohr [1] examined a wide variety of interest point detectors and categorized them, based on their localization criteria, into three main groups: Contour-based, Intensity-based and Parametric-model based methods. The Contour-based methods define interest points either at the intersections of grouped line segments or at the maximum curvature of approximated contours. Intensity-based methods define interest points through the illumination distribution of the neighborhood. In most cases these algorithms are based on the second moment matrix, which is a mathematical measure for the distribution of the local image gradients. Parametric-based methods on the other hand define interest points at regions that fit a predefined analytical intensity model. This paper focuses on a group of Intensity-based detectors [3, 4], which define the interest points as the local peaks of grayvalue derivatives in scale-space. In most cases these detectors are capable of identifying local patterns independent from any scale changes. In this paper we propose a new non-linear

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 643 – 652, 2006.

scale-space representation, which improves the localization accuracy of the aforementioned detectors [3, 4].

In all our experiments we used the SIFT descriptor [3] to define the local patterns at each interest point. Mikolajczyk and Schmid [7] proved that the SIFT descriptor is more robust to affine changes than many other descriptors including steerable filters [8], differential invariants [2, 9], complex filters [11] and moment invariants [10]. We did also use a modified version of the SIFT descriptor which is more distinctive and in many cases leads to a much better matching results.

Overview. Section 2 presents different implementations for the scale-space including a new proposal, which in general uses the non-linear spatial filter of Köthe [6]. Section 3 reviews the main features of the detectors and descriptors used in our tests. Section 4 introduces the evaluation criteria. Section 5 and 6 present the experimental results and the conclusion.

2 Scale-Space Representations

A linear scale-space is defined by the solution of the following diffusion equation;

$$\frac{\partial L(z,s)}{\partial s} = \frac{1}{2}\nabla^2 L(z,s) = \frac{\partial_{xx}L(z,s) + \partial_{yy}L(z,s)}{2} \tag{1}$$

with the initial condition that $L(z,0)$ (i.e. initial scale s=0) is equal to the original image $I(z)$, ∇^2 is the Laplacian kernel and z is the spatial coordinates of the interest point. Equivalently a linear scale-space can be defined by convolving $I(z)$ with the Gaussian kernel G(z,s).

$$G(z,s) = \frac{1}{2\pi\sqrt{s}} e^{-z^2/2s^2} \tag{2}$$

To reduce the amount of smoothing around edges Perona and Malik [5] proposed the use of anisotropic diffusion as a generalization of the linear scale-space representation.

$$\frac{\partial L(z,s)}{\partial s} = \frac{1}{2}\nabla^2 \left(h(z,s)\ \nabla L(z,s)\right) \tag{3}$$

where $h(z, s)$ is defined to be dependant on the image gradient. A possible solution for $h(z, s)$ is presented by eq.4 where k defines the range of gradients in an image and thus controls the amount of smoothing at point z.

$$h(z,s) = e^{-\frac{|\nabla L(z,s)|}{K}} . \tag{4}$$

2.1 Hourglass Representation

Köthe [6] proposed an oriented non-linear spatial filter that looks like an hourglass. The new filter modulates the Gaussian so that it becomes zero at a perpendicular distance from the local edge direction ϕ_0. The output of the filter at point (x,y) is given by the following equation:

$$h_{\sigma,\rho}(z,\phi,\phi_0) = \frac{1}{N} e^{-\frac{z^2}{2\sigma^2}} e^{-\frac{\tan^2(\phi-\phi_0)}{2\rho^2}} \tag{5}$$

where z and ϕ are the polar coordinates of point (x, y), ρ defines the width of the Hourglass filter, the larger the value of ρ the more the filter tends to become uniform, and N is a normalization factor that sums the weights of the filter to 1. Köthe recommended that ρ should be set to a value between 0.3 and 0.7.

The dimension of the Hourglass scale-space is defined by an initial scale σ_0, final scale σ_f, and a factor *k* of scale change between successive levels. At each scale level σ a local direction ϕ_o is calculated for each sample point using a simple derivative function. Next the Hourglass kernel is rotated by ϕ_o degrees and applied to the sample point.

3 Experiment Setup

In the following we will review the implementation details of two interest point detectors and two descriptors used in our experimental tests. The detectors are invariant to scale and rotation changes. The descriptors on the other hand are distinctive and relatively robust to common image transformations.

3.1 Interest Point Detectors

The detection scheme in the following two algorithms starts with an appropriate implementation of the scale-space.

SIFT: first, local peaks are selected from a Difference of Gaussian pyramid. A 3D quadratic function is fitted at each local peak and an interest point location is calculated up to a sub-pixel /sub-scale accuracy at the extremum value of this quadratic function. Finally interest points with low contrast values and points located along edges are considered unstable and rejected.

Harris-Laplacian [4]: a scale-space is built for the Harris function using the second moment matrix C(z,s,s^{-}). At each scale-space level s the local peaks of the Harris function are selected as possible interest point candidates. Finally, candidates with the local scale-space maximum of the Laplacian function are identified as interest points.

Harris function = det(C) - αtrace^2(C

$$\textit{Where} \quad C(z,s,s^{-}) = s^2 G(z,s^{-}) * \begin{bmatrix} L_x^2(z,s) & L_x L_y(z,s) \\ L_x L_y(z,s) & L_y^2(z,s) \end{bmatrix}, \tag{6}$$

L_z and L_y are the gradients along the x and y axis respectively.

3.2 Descriptors

The descriptors used in our tests are: (1) the standard SIFT and (2) a modified version of the SIFT. In the remaining part of this section we will review the design aspects of these two descriptors.

SIFT: A descriptor is calculated for each interest point with a spatial location z and scale s through to the following steps:

1. A dominant orientation angle θ is calculated from the local neighborhood of p, which is defined by a circular region of radius 1.5s. The method of detecting θ is explained in detail in [3].
2. A local window W of size 16x16 is fitted at location z and scale s.
3. A gradient orientation and magnitude are calculated for each sample point that lies within W.
4. To achieve rotation invariance, the coordinates and the gradient orientations of W are rotated by angle $-\theta$.
5. The gradient magnitudes of W are smoothed with a uniform Gaussian kernel of scale k=1.5 the width of W. This step is meant to reduce the effect of sample points that lie away from z as they are considered the most likely affected points with misregistered errors.
6. The local window W is divided into 16 different 4x4 sample regions.
7. The weighted gradient magnitudes of each sample region are summed in an orientation histogram with eight directions as shown in figure.1.

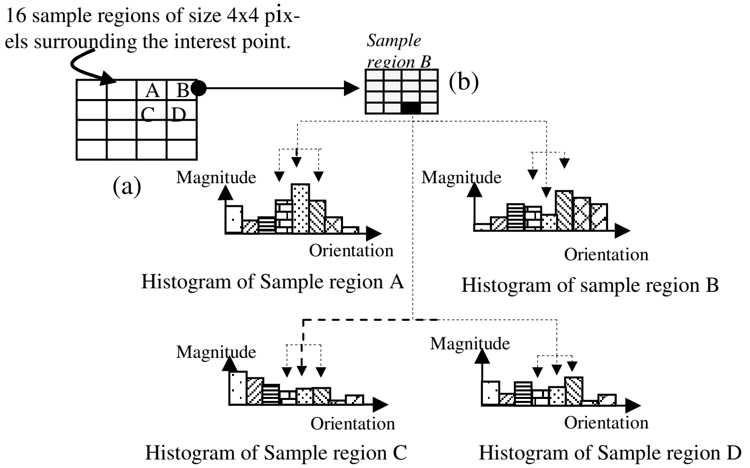

16 sample regions of size 4x4 pixels surrounding the interest point.
A
B
C
D
(a)
Sample region B
(b)
Magnitude
Orientation
Histogram of Sample region A
Magnitude
Orientation
Histogram of sample region B
Magnitude
Orientation
Histogram of Sample region C
Magnitude
Orientation
Histogram of Sample region D

Fig. 1. (a) The neighborhood of the interest point is divided into 16 sample regions. (b) The gradients of each sample region (i.e. as in region B) are accumulated in an orientation histogram with 8 directions and distributed among the histogram bins of neighboring regions (i.e. regions A, C and D) through a tri-linear interpolation.

8. The descriptor is formed from a vector containing the values of all the 8x16=128 orientation histogram bins.
9. To reduce the effects of illumination change the vector elements are normalized to a unit length, then thresholded to values not greater than 0.2 and finally renormalized.

Modified-SIFT: Steps '1' and '5' in the above algorithm are modified and applied for each interest point z with scale s as follows:

- *Step 1*: In the SIFT algorithm the pixels at spatial distances less than 1.5*s* from *z* are defined as the local neighbors of *z* while in the modified-SIFT the pixels with both grayvalue and spatial distances less than 1.5*s* are defined as the local neighbors of *z*.
- *Step 2*: A Gaussian function with scale *k* is used to weight the gradients of the local neighbors of point *z* in the SIFT algorithm. The weight is set to decrease exponentially as the spatial distance between the local neighbor and point *z* increases. In the modified-SIFT a weight $w_i(c)$ is assigned for each local point *i* using the function of equation.7. The weight $w_i(c)$ is defined in terms of the gravalue distance *c* between *i* and point *z*.

$$w_i(c) = \frac{1}{2\pi\sqrt{k}} e^{-c^2/2k^2} \tag{7}$$

The reason behind the above modifications is that normally local regions are identified by their color distribution. The distribution is in most cases continuous and of size proportional to the scale of the local region.

4 Evaluation

We have conducted two matching tests to measure the performance of the interest point detectors of section 3.1 before and after applying the Hourglass scale-space representation and the SIFT descriptor before and after applying the modifications of section 3.2.

In the first test a number of synthetically transformed images were used for matching. These transformations included, scale changes, rotation, brightness changes and noise addition. In this test the Receiver Operating Characteristics (ROC) curves were used for evaluation as indicated by Carneiro and Jepson [12], where for each type of transformation and each feasible combination of the three different elements under test (i.e. scale-space representation, interest point detector and local descriptor) a detection rate versus a false positive rate is plotted.

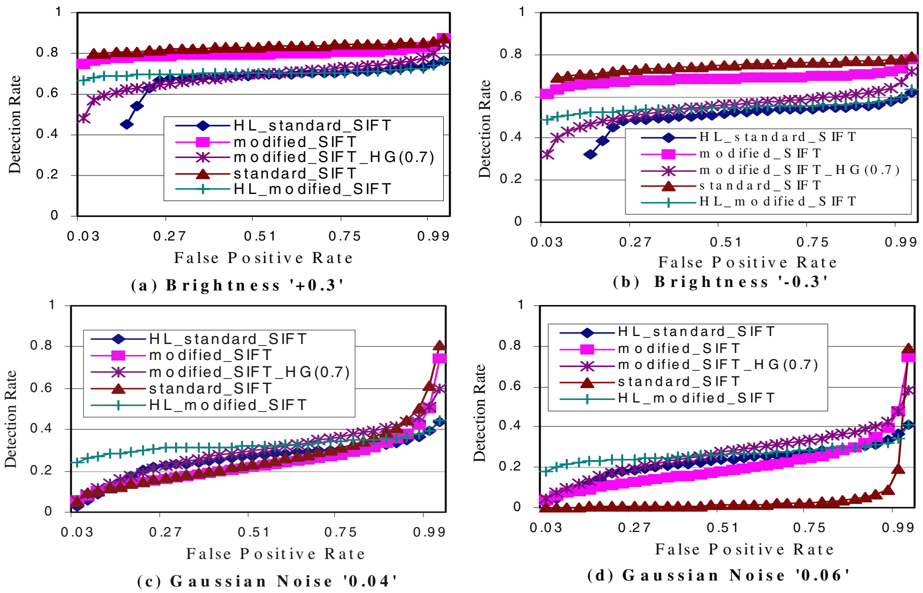
Given a test image **I** and its transformed version $\mathbf{I}'$, where **I'= M I+b**, a detection rate is defined as the ratio between the number of correct matches (correct-positives) and the total number of interest points of **I**. A correct match is scored between two interest points **x** and **y**, where $\mathbf{x} \in \mathbf{I}$ and $\mathbf{y} \in \mathbf{I}'$, if **y** is very close to the mapped point **x'=M x+b** (i.e. $\|\mathbf{y}-\mathbf{x}'\| < \varepsilon$) and has nearly the same local descriptor as **x** (i.e. $\|D(\mathbf{y})-D(\mathbf{x})\| < \tau$).

On the other hand given a database of images that doesn't include **I** nor **I'**, a false positive rate is defined as the ratio between the number of false matches (false positives) and the total number of interest points of **I**. A false match is scored if there exists an interest point **z** in the database that is similar to **x** (i.e. $\|D(\mathbf{z})-D(\mathbf{x})\|< \tau$). In our tests ε was set to 3 pixels and τ was changed in regular steps of 0.03 to form the ROC curves.

The second test involved matching real images taken from different viewpoints. In this test the evaluation of the matching results of each image pair (**I**, **I'**) was based on the following criteria: for each interest point **x** that belongs to **I** the two points ($\mathbf{x_1}$ and $\mathbf{x_2}$) with the most similar descriptors to **x** are identified in **I'**, where $\|D(\mathbf{x_1})-D(\mathbf{x})\|< \|D(\mathbf{x_2})-D(\mathbf{x})\|$. Next $\mathbf{x_1}$ is considered a valid match to **x** if $\|D(\mathbf{x_1})-D(\mathbf{x})\|$ is less than 90% of $\|D(\mathbf{x_2})-D(\mathbf{x})\|$. For further validation the matching results of this test were visually inspected and reported in table.3.

5 Results

The 8 test images of figure.4.a and a database of 60 different images representing a collection of natural scenes were used to create the ROC curves of figure 2, 3 and 5. These curves were designed to evaluate the performance of the five different techniques of table.1. In this test a total of 1.04 million interest points were detected according to the distributions of table.2.



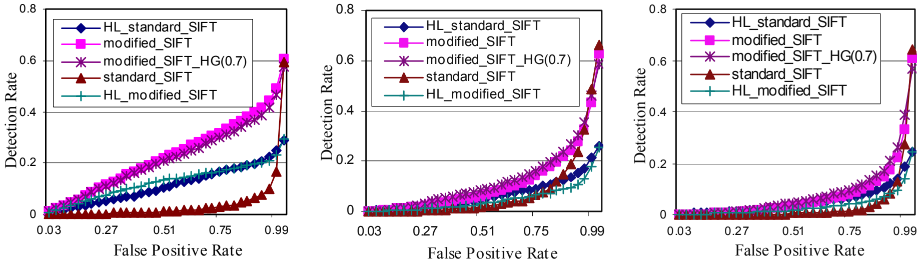
(a) Brightness '+0.3'

(b) Brightness '-0.3'

(c) Gaussian Noise '0.04'

(d) Gaussian Noise '0.06'

Fig. 2. ROC curves for simple image transformations that include (a) an increase in the illumination by a factor of 0.3 and (b) a decrease in the illumination by a factor of 0.3, and an addition of Gaussian noise with variances of (c) 0.04 and (d) 0.06. The curves were plotted for interest points detected by the SIFT and the Harris_Laplacian(HL) detectors and matched through the SIFT and *modified_*SIFT descriptors.



(a) Rotation '+15' (b) Rotation '+30' (c) Rotation '+45'

Fig. 3. ROC curves for image rotations of 15, 30 and 45 degrees

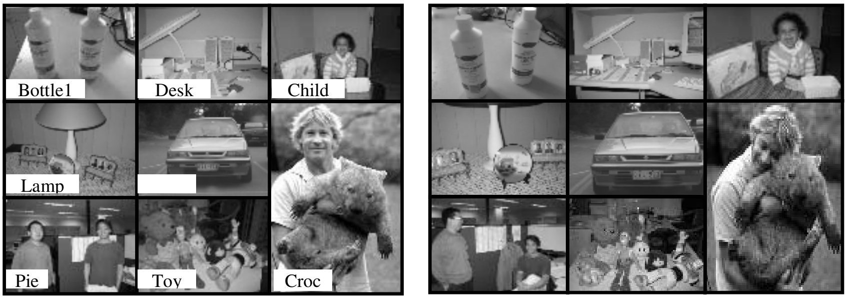


Fig. 4. Test images including the (a) original series and (b) an affine-transformed version

Table 1. The five techniques under test

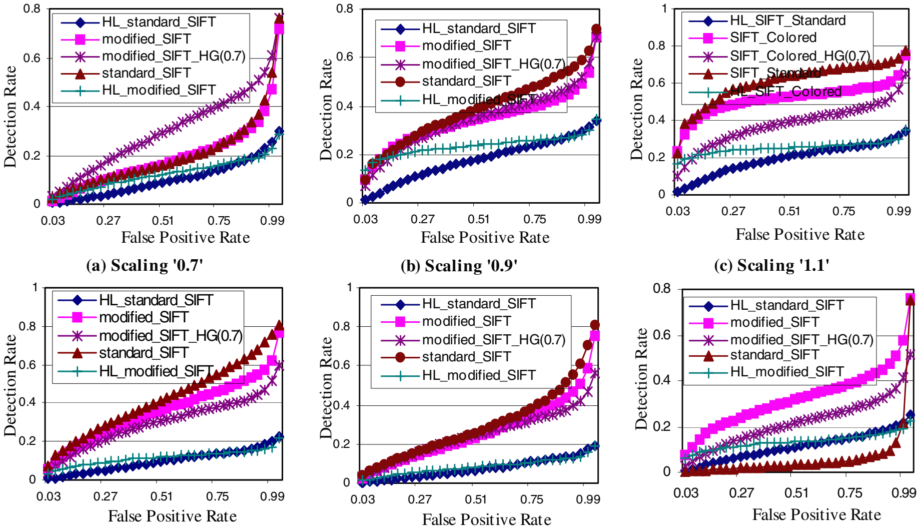
Method Title	Detector	Descriptor	Scale-Space
HL_*standard*_SIFT	Harris Laplacian	SIFT	Linear
*modified*_SIFT	SIFT	*modified*_SIFT	Linear
*modified*_SIFT_HG(0.7)	SIFT	modified_SIFT	Hourglass $\rho=0.7$
*standard*_SIFT	SIFT	SIFT	Linear
HL_*modified*_SIFT	Harris Laplacian	*modified*_SIFT	Linear

In case of the Hourglass scale-space, experimental results showed that the number of detected interest points is directly proportional to the size of the smoothing kernel and inversely proportional to the value of the ρ-parameter (see equation.5), where in general an increase of 0.2 in the value of ρ results in the reduction of the number of points by a factor of 0.81. Making use of this fact and in order to speed up the process of building the Hourglass scale-space the SIFT algorithm was slightly modified, where instead of expanding the input image by a factor of 2 the first level of the Gaussian pyramid was sampled at the same rate of the input image and the smoothing kernel was increased from size 7 to 13. This automatically implies that in case of the Hourglass scale-space no interest points can be detected with a scale less than 0.5.

The ROC curves of figures 2a and 2b show that under illumination changes the highest two detection rates were scored for the *standard*_SIFT and the *modified*_SIFT consequently. The HL_*modified*_SIFT was ranked third up to a false positive rate of 0.27. At false positive rates greater than 0.27 the *modified*_SIFT_HG was ranked third and both the HL_*modified*_SIFT and the HL_*standard*_SIFT were ranked fourth.

The curves of figures 2c and 2d show that the HL_*modified*_SIFT is the most resistant to noise at lower false positive rates while the *modified*_SIFT_HG performs much better at higher false positive rates.

To evaluate the performance for orientation changes the test images were rotated at 15, 30 and 45 degrees and the ROC curves were plotted for each angle change. The results of figure 3 show that the *modified*_SIFT and the *modified*_SIFT_HG worked much better than the other three techniques for all the three angle changes with an exceptional performance at angle 15.



(a) Scaling '0.7' **(b) Scaling '0.9'** **(c) Scaling '1.1'**

(d) Scaling '1.3' **(e) Scaling '1.5'** **(f) Scaling '1.8'**

Fig. 5. ROC evaluation curves for scale changes between 0.7 and 1.8

The matching results of figure 5 involve a wide range of scale changes starting from a factor *f* of 0.7 and increasing in steps of 0.2 up to a factor of 1.8. The ROC curves show that the *modified*_SIFT_HG performed outstandingly well at *f*=0.7, the *standard*_SIFT dominated the range between 0.9 and 1.5 and the *modified*_SIFT had the highest detection rates at *f*=1.8. Moreover in the range between 0.9 and 1.1 the HL_*modified*_SIFT worked much better than the HL_*standard*_SIFT.

The reason behind the results of figure 5.a is that in the linearly smoothed version of a downscaled image the nearby edges merge causing small structures to disappear and consequently affects the localization accuracy of the interest points. On the contrary the *modified_*SIFT_HG preserves these structures through non-linear smoothing, which in turn lead to a more accurate localization and much better matching results. Moreover the inadequate performance of the *modified_*SIFT_HG at f >1 (i.e. see figures 5c - 5.f) was due to the fact that the *modified_*SIFT_HG usually ignores the local structures of very high spatial frequencies (i.e. scales less than 0.5) and in turn reduces the number of valid matches between the input image and its scaled version.

The results of figure 5.f show that the *modified*-SIFT descriptor is more robust to large scale changes than the *standard*-SIFT because it gives more emphasis to local neighbors with similar gray values to the interest point and consequently is affected by less misregistration errors. The matching results of table.3 further prove that the modified_SIFT_HG algorithm is more resistant to affine changes than the standard_SIFT algorithm.

Table 2. Distribution of the detected interest points

Image Group	%	Method	%
Image Database	41	HL_*standard*_SIFT	13
Test Images	4	*modified*_SIFT	25
Transformed Test Images	55	*modified*_SIFT_HG(0.7)	26
		*standard*_SIFT	20
		HL_*modified*_SIFT	16

Table 3. Visually inspected matching results for the test images of figures 4.a and 4.b

Image Title	Percentage of valid matches		
	standard_SIFT	modified_SIFT	modified_SIFT_HG (0.5)
Bottle	2.72	7.09	19.9
Child	5.36	13	38.1
Croc	5.88	16.8	18
Desk	8.14	17	36.6
Lamp	0.623	2.2	12.3
Pei	2.71	7.25	13.6
Toy	9.7	13.7	24.6
Car	12.8	24.9	44.8

6 Conclusion

In this paper we have presented an experimental evaluation for a new non-linear scale-space representation and a modified version of the SIFT descriptor. The evaluation was based on matching images with both synthetic and real geometric transformations. Two different techniques were used for evaluation including the Receiver Operating Characteristic (ROC) curves and an ordinary visual inspection method. The standard SIFT descriptor proved to have better matching results under illumination changes. The results of the proposed non-linear scale-space and the *modified*_SIFT descriptor were superior under orientation and large-scale changes.

The assumption of eliminating the local structures of very high spatial frequencies from the proposed non-linear scale-space proved to be a time saving step. On the other hand it underestimated the matching results of the *modified*_SIFT descriptor.

References

1. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of Interest Point Detectors. International Journal of Computer Vision, Vol. 37, Issue 2. (2000) 151–172.
2. Schmid, C., Mohr, R.: Local gray value invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, Issue 5. (1997) 530–535.
3. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. (2004).
4. Mikolajczyk, K., Shmid, C.: Indexing based on scale invariant interest points. Proceedings of the 8th International Conference on Computer Vision. Vancouver, Can., (2001) 525-531.
5. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, Issue 7. (1990) 629-639.
6. Kothe, U.: Edge and Junction Detection with an Improved Structure Tensor. The 25thDAGM Symposium Mustererkennung. LNCS, Vol. 278. Springer (2003) 25-32.
7. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. Proceedings of Computer Vision and Pattern Recognition. (2003).
8. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, Issue 9. (1991) 891–906.
9. Koenderink, J., van Doorn, A.: Representation of local geometry in the visual system. Biological Cybernetics, Vol. 55. (1987) 367–375.
10. van Gool, L., Moons, T., Ungureanu, D.: Affine/photometric invariants for planar intensity patterns. Proceedings of European Conference on Computer Vision. (1996).
11. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets. Proceedings of European Conference on Computer Vision, Vol. 1. (2002) 414–431.
12. Carneiro, G., Jepson, A.D.: Phase-based Local Features. Proceedings of European Conference on Computer Vision, Vol. 1. (2002) 282–296.

Feature Selection for Image Categorization

Feng Xu and Yu-Jin Zhang

Department of Electronic Engineering, Tsinghua University,
Beijing 100084, China
f-xu02@mails.tsinghua.edu.cn, zhang-yj@mail.tsinghua.edu.cn

Abstract. Image classification could be treated as an effective solution to enable keyword-based semantic image retrieval, while feature selection is a key issue in categorization. In this paper, we propose a novel strategy by using feature selection in learning semantic concepts of image categories. To choose representative and informative features for an image category and meanwhile reduce noisy features, a feature selection strategy is proposed. In the feature selection stage, salient patches are first detected by SIFT descriptor and clustered by DENCLUE algorithm. Then the pointwise mutual information between the salient patches and the image category is calculated to evaluate the important patches and construct the visual vocabulary for the category. Based on the selected visual features, the SVM classifier is applied to categorization. The experimental results on Corel image database demonstrate that the proposed feature selection approach is very effective in image classification and visual concept learning.

1 Introduction

CBIR is a challenging task for a large-scale image database and web pages due to the semantic gap between low level features and high level semantic concepts. An alternative solution is to search images by text keywords, which makes the automated or semi-automated image categorization and annotation increasingly important. A successful annotation and categorization will significantly enhance the performance of content-based image retrieval systems by filtering out images from irrelevant classes during matching.

Many good results have been reported in two class image classification tasks, such as city vs. landscape [1], indoor vs. outdoor [2]. Recently, many promising approaches for general object recognition were proposed and demonstrated to solve multiple class image classification tasks. Fergus et al. proposed constellation model, which is learned in a Bayesian manner, to recognize six classes of objects [3]. The model could be learned from unlabeled and unsegmented cluttered scenes in a scale invariant manner, and is capable of recognizing six object classes. This classification scheme was further improved by Li et al. to classify more categories with less training samples [4]. A good application of this scheme is filtering Google images [5]. Taking into account shape, appearance, occlusion and relative scale, the constellation model well describes an object in multiple

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 653–662, 2006.

semantic aspects with low-level features, and demonstrates promising potentials in image understanding. However, its computational cost is too expensive in both learning and recognition, and it is difficult to extend the algorithm to large-scale image databases. Csurka et al. proposed bags of key-points of objects as features. Based on that, the visual vocabulary is constructed by k-means clustering algorithm. Both Naïve Bayes and SVM classifiers are applied to categorization [6]. But the noise affected the results significantly.

On the other hand, it is useful to have access to high-level information about objects contained in images to manage image collections. To achieve this goal, high-level information must be learned and modeled from low-level features. A significant number of models have been proposed to model objects from low-level features. As low-level features are usually noisy and uninformative, feature selection is of great importance and needs to be conducted before modeling object. Nuno et al. [7] exploited recent connections between theoretic feature selection and minimum Bayesian error solutions to derive feature selection algorithm that are optimal in a discriminant feature sense without compromising scalability. However, they did not provide feature selection from image content, in which semantic feature is not included.

In image classification, features are required to be representative within the same class and discriminative for different classes. Therefore it is essential to select the most informative features. In text categorization, a significant number of feature selections have been proposed in order to reduce the dimensions of the documents. In image categorization, feature selection should be proposed to extract more effective features. Thus a robust feature selection strategy based on image category is crucial and worthy of investigation. Based on the selected features, images can be expressed as combination of the informative visual keywords.

In this paper, we propose a novel image classification framework. First, a feature selection strategy is explicitly conducted. For every image category, the salient patches on each image are detected by SIFT (Scale Invariant Feature Transform [8, 9]) and quantized by DENCLUE (DENsity-based CLUstEring) [10] algorithm. Then a visual keyword dictionary is constructed. Unlike the method in [6], the proposed visual keywords collection is coutinuous while the visual vocabulary in [6] is discrete (i.e. only the cluster centers are applied). Thereafter, the pointwise mutual information between each salient patch and image category is calculated. Since the clustering is conducted as the patch distribution density, those salient patches with larger pointwise mutual information to the category are selected. The larger the distribution density is, the more the salient patches are selected. Based on the selected patches, the categorization is performed. The SVM classifier, as the widely used discriminative classification model, is applied. Compared with other image classification methods, we focus on the effectiveness of the proposed feature selection algorithm.

The rest of the paper is organized as follow: Section 2 presents the extraction of the salient patches; section 3 proposes feature selection strategy; Section 4 presents the classification methods, SVM classifier and the utility of the features;

Section 5 shows the experimental results to evaluate the performance of our techniques; Section 6 gives conclusions.

2 Salient Patches Extraction

Recent progresses in object recognition and image annotation have shown that local salient features are more informative in describing image content than global features [3, 4, 6]. So the local salient features are applied in the proposed method. The salient patches extraction includes two steps, detection and description.

2.1 Salient Patches Detection

Object recognition in cluttered real-world scenes requires local image features that are unaffected by nearby clutter or partial occlusion. The features must be at least partially invariant to illumination and sufficiently distinctive to identify specific objects among many alternatives. So local features are preferred. SIFT is developed by Lowe [8, 9]. In [11, 12] this descriptor was shown to be superior to others used in the literature. Therefore, SIFT is used to detect the local features in the proposed approach.

SIFT is built by selecting key locations at maxima and minima of a difference of Gaussian function applied in scale space. The local maxima and minima are not only in the same level, but also in the adjacent level of the Gaussian pyramid. It can identify location in image scale space that are invariant with respect to image translation, scaling, and rotation, and are minimally affected by noise and small distortions. It generates large numbers of features that densely cover the image over the full range of scales and locations. By SIFT descriptor, image data are transformed into scale-invariant coordinates relative to local features. Generally, this detector gives hundreds of salient patches for a typical 256 by 384 (or 384 by 256) pixel image, without color information included.

2.2 Salient Patches Description

Once the salient patches are identified, they are cropped from the image and rescaled to the size of a small pixel patch. Because a high dimensional description is difficult to manage, principal component analysis (PCA) is performed on the patches from all images. Then each patch is represented by a vector of the coordinates within the first 10 principal components. Thus each salient patch is described as a 10-dimensional feature vector.

3 Feature Selection Strategy

Feature selection aims at the most informative and discriminative features of the image category. From the point of view of information theory, whether a feature is informative can be well evaluated by pointwise mutual information between the feature and the class. Pointwise mutual information has been proved to be effective in text classification [13]. Since the detected local salient patches can be clustered as 'visual keywords', the pointwise mutual information between salient patches and image category can be used to select informative features.

3.1 Salient Patches Clustering

The 10-dimensional PCA salient features are clustered to construct visual keyword dictionary. The DENCLUE clustering algorithm is applied.

It is assumed that $f^i(x)$ is the influence function of a data object. The density function is defined as the sum of the influence functions of all data points. Given N data objects described by a set of feature vectors $D = \{x_1, \cdots x_N\}$, the density function is defined as

$$f(x) = \sum_{i=1}^{N} f^{x_i}(x) \tag{1}$$

In principle, the influence function can be an arbitrary function. Here, two types of influence functions are applied and Euclidean distance is used.

1. Distance influence Function:

$$f(x, y) = \frac{1}{d(x, y)} \tag{2}$$

The corresponding density function is

$$f(x) = \sum_{i=1}^{N} \frac{1}{d(x, x_i)} \tag{3}$$

2. Gaussian influence Function:

$$f(x, y) = e^{-\frac{d(x,y)^2}{2\sigma^2}} \tag{4}$$

The corresponding density function is

$$f(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\sigma^2}} \tag{5}$$

This function is affected by the parameter σ significantly.

The details about DENCLUE can be referred to [10].

Thus, the density of each data in each cluster can be estimated. After obtaining the density of the whole data set, a Parzen estimation algorithm is applied to obtain the clusters [14]. Assume that the current partition with density function $f(x_i \mid C_j)$ for each data x_i in each cluster $C_j, j = 1, \cdots J$. The objective function is:

$$f(x_i \mid C_j) = \max_l f(x_i \mid C_l) \tag{6}$$

So three steps are conducted in this clustering algorithm:

Step 1: Initializing clusters for the data set.
Step 2: For each data x_i, conditional densities in each cluster are calculated. Then label x_i according to Eq. (6).
Step 3: If some labels of x_i are changed, go to step 2.

Thus the data are clustered according to the distribution density. The higher the density is, the compact the cluster is. The clusters of salient patches can be regarded as a continuous visual keyword dictionary.

The advantages of DENCLUE are mainly in: it has a firm mathematical basis; it has good clustering properties in data sets with large amounts of noise; it allows a compact mathematical description of arbitrarily shaped cluster in high-dimensional data sets.

3.2 Pointwise Mutual Information

The estimated density function can be regarded as probability density function after normalization.

$$p\left(\vec{x_i} \mid C_j\right) \propto f\left(\vec{x_i} \mid C_j\right) \tag{7}$$

where $p\left(\vec{x_i} \mid C_j\right)$ is the estimated class conditional probability density function.

When the class conditional probability density is estimated with the clustering, the pointwise mutual information of a salient patch in each cluster can be calculated. The pointwise mutual information between the salient patch $\vec{x_i}$ and the class C_j is:

$$I\left(\vec{x_i}, C_j\right) = \log \frac{p\left(\vec{x_i} \mid C_j\right)}{p\left(\vec{x_i}\right)} \tag{8}$$

where $p\left(\vec{x_i}\right)$ is the estimated probability of salient patch in the image category.

Pointwise mutual information tells us how information the occurrence of one visual keyword is about the occurrence of one cluster. If the pointwise mutual information of a visual keyword in a cluster is high, this visual keyword contributes and influences more to the cluster.

3.3 Representative Salient Patches Selection

The pointwise mutual information between the salient patch and the data clusters reflects how a salient patch is representative in the image category. So the pointwise mutual information between the salient patch and the data cluster can be regarded as the pointwise mutual information between the salient patch and the image category. Therefore, those salient patches with higher pointwise mutual information in image category are selected as the features for this image category.

All the salient patches are ranked as the pointwise mutual information ranging from high to low and the top M salient patches are selected. M is the pre-determined number of features. M should be balanced between the computation complexity and the capability of description. Too many salient patches will lead to large quantity of computation while too few salient patches will lead to incomprehensive description for image content.

The clusters by DENCLUE are obtained according to density of data distribution. In order to avoid the selected features with the higher pointwise mutual information are from the same cluster, the density is used as weight. For the cluster with higher density, more salient patches are selected; for the cluster with lower density, less salient patches are selected.

Thus, the selected salient patches represent not only the relation to the image category but also the feature distribution. These selected salient patches can describe the image category comprehensively.

4 Image Categorization

Once the feature descriptors have been selected, the problem of generic visual categorization is reduced to that of multi-class supervised learning. As SVM (Support Vector Machine) is a well-known classifier to produce state-of-the-art results in high-dimensional problems, we apply SVM classifier to the image classification.

SVM classifier aims at finding a hyperplane which separates two-class data with maximal margin [15]. The margin is defined as the distance of the closest training point to the separating hyperplane. For given observations $\vec{x}$, and the corresponding labels $\vec{y}$ which take values ±1, SVM will find a classification function:

$$f\left(\vec{x}\right) = sign\left(\vec{w^T}\vec{x} + \vec{b}\right) \tag{9}$$

where $\vec{w}$ and $\vec{b}$ are the parameters of the classifying plane.

In the visual categorization task, $\vec{x}$ is the selected feature vector, in which the representative salient patches from each visual keyword in an image category are integrated into training set. The top M salient patches with the higher pointwise mutual information are selected, in which M is a pre-determined number. The similarity measure is the combination of the most informative salient patches in the classification. The elements of the selected feature vector with the higher pointwise mutual information are used to measure the similarity between two images. In order to apply SVM to multi-class problems we take the one-against-all approach. That is, given an m-class problem, we train m SVM classifiers. Each classifier distinguishes images in one category from all the other m-1 categories. Given a query image, the salient patches are also extracted and classified by the SVM classifier. Then the label frequencies of all the salient patches are counted. Finally the label with the highest frequency of the salient patches is assigned to the image.

5 Experimental Results and Discussions

The experiments are conducted on Corel image database. The 25 image categories with labels corresponding to object semantic concepts are selected. Each image category consists of 100 images, in which 80 images are used to train the classifier and the other 20 images are used to test. To make experiments more convincing, a 5-fold cross validation has been carried out.

The benchmark metrics for classification evaluation are classification *precision* α and *recall* β, defined as:

$$\alpha = \frac{\phi}{\phi + \varepsilon}, \beta = \frac{\phi}{\phi + \eta}$$

where ϕ is the number of true positive samples that are correctly classified to their corresponding category, ε is the number of true negative samples that are irrelevant to the corresponding category and are classified incorrectly, η is the number of false positive samples that are related to the corresponding category but are misclassified.

In SVM classification, each category is classified by linear binary classifier. Using kernel SVM will possibly improve the performance, but here we compare the performance between methods with feature selection and without feature selection, in which linear SVM performs well. The features before feature selection and after feature selection are used respectively. In our experiment, 100 salient patches per image can describe image comprehensively. So M is set to 100. For SVM classifier with feature selection, the salient patches on each image are arranged as the pointwise mutual information and the top 100 patches are integrated as the training data, weighted by the cluster density. For SVM classifier without feature selection, the 100 salient patches are selected randomly.

The performance of the 5-fold cross validation is shown in Table 1.

Table 1. The precisions and recalls of 5-fold cross validation

	Precision	Recall
Validation 1	69.03%	57.90%
Validation 2	63.51%	58.42%
Validation 3	69.00%	55.87%
Validation 4	73.83%	62.83%
Validation 5	61.94%	55.06%
Average	67.46%	58.02%

From this table, the performance of the proposed feature selection and classification method has been exhibited to be effective.

In one of the classifications, the precisions of all image categories are illustrated in Fig. 1.

From this figure, it can be found that most of the image categories can achieve precisions over 50%. Those categories with concrete concept, such as *Building*, *Bus*, *Firework* etc, achieve quite higher precisions. However, there are also some categories perform weakly, such as *Beach* and *Ski*, which is probably due to the diversity and complex background and noise.

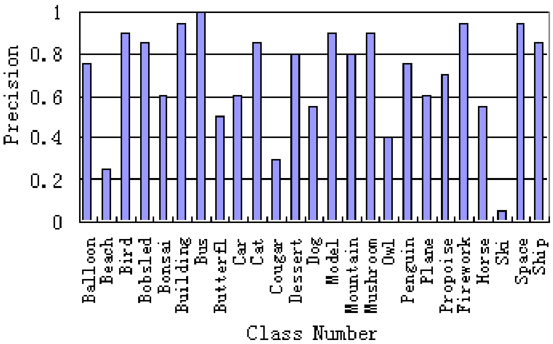


Fig. 1. Precisions of image categories in Validation 1

Table 2. Comparison between precisions without feature selection and with feature selection

	Without feature selection	With feature selection
Average Precision	56.67%	67.46%

Comparison between the precisions without feature selection and with feature selection is shown in Table 2.

Compared with precision without feature selection, the precision with feature selection has been improved over 10 percent, which proves the effectiveness of the proposed approach. The most representative and informative features are selected through pointwise mutual information and the image categories can be classified discriminatively.

Although the visual keyword method is similar to that reported in [6], the results cannot be directly compared with each other due to the different image database. However, only seven image categories are used in the experiment in [6] while dozens of image categories are used in our experiment. Some precisions in Fig. 1 suggest that our approach will give as good results as that in [6].

The continuous visual vocabulary is more appropriate to image categorization since the image features are in continuous space instead of the discrete feature space of the text. If the image feature space is quantized and only the cluster centers are applied as the discrete visual keywords, it is probably lose some important image information. So the feature selection according to the cluster density and pointwise mutual information is more reasonable and effective. Fig. 2 shows several images in the same category, in which the images with all the detected SIFT descriptors are shown in the first row and the images with the selected descriptors are shown in the second row.

From the above images, it can be found that the representative salient patches are preserved by feature selection. In *Beach* category, a majority of preserved points are located on *sand*, *sky* and *sea* which are the most relevant to the category concept. Secondly, the pointwise mutual information promises that the most

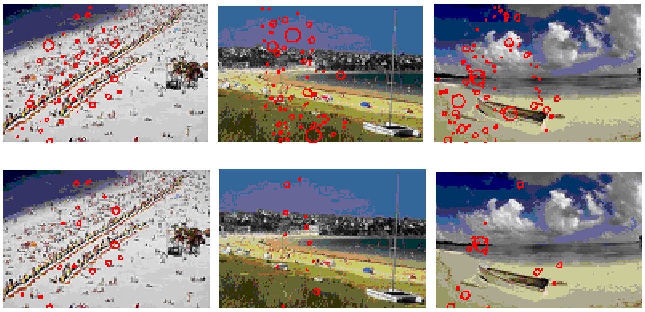


Fig. 2. Image examples from *Beach* category. The upper images show all the detected patches and the lower images show the selected patches.

informative points are preserved. Thirdly, the continuous visual vocabulary can be explained by an example in Fig. 3. The left image illustrates all the detected patches correspoingding to *window* concept while the right image illustrates the selected patches. In the *Building* image, the salient patches on windows tend to be clustered together. Several salient patches located on windows are selected in the proposed approach while only one key point (cluster center) is selected in discrete visual keyword [6]. Since *window* is one of the most important features for *building* concept, the corresponding cluster is always with higher distribution density. So the clustering according to density is more effective.

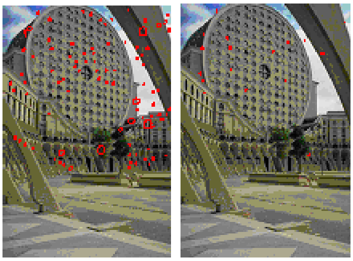


Fig. 3. An image example with patch cluster corresponding to *window*. The left image shows all the patches while the right image shows the selected patches.

6 Conclusions and Future Work

In this paper, a novel content-based feature selection approach is proposed for image classification. To select features for classification, the salient patches are detected by SIFT and the 10-dimentsional feature vectors are formed by PCA. Then the EDNCLUE clustering algorithm is applied to construct the continuous visual keyword dictionary. After estimating the density, the pointwise mutual information between the salient patches and the class is calculated and used to select the representative patches. Finally, the SVM is used for classification. The experimental results prove the effectiveness of the proposed approach.

In the future, unsupervised feature selection algorithm should be investigated. The feature selection can also be applied to image annotation.

Acknowledgement

This work has been supported by the Grants NSFC-60172025 and SRFDP-20050003013.

References

1. Aditya Vailaya, Anil Jain, Hong Jiang Zhang. On Image Classification: City vs. Landscape. Pattern Recognition, 31(12):1921-1935, 1998.
2. M. Szummer, R. Picard. Indoor-outdoor image classification. IEEE International Workshop on Content-based Access of Image and Video Databases, 42-51, 1998.
3. R. Fergus, P. Perona, A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. *CVPR*, 2:264-271, 2003.
4. Li Fei-Fei, Rob Fergus, Pietro Perona. A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories. *ICCV*, 2:1134-1141, 2003.
5. R. Fergus, P. Perona, A. Zisserman. A Visual Category Filter for Google Images. *ECCV*, 242-256, 2004.
6. Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, Cedric Bray. Visual Categorization with Bags of Keypoints. *ECCV*, 11-14, 2004.
7. Nuno Vasconcelos, Manuela Vasconcelos. Scalable Discriminant Feature Selection for Image Retrieval and Recognition. *CVPR*, 2:770-775, 2004.
8. David G. Lowe. Object Recognition from Local Scale-Invariant Features. *ICCV*, 2:1150-1157, 1999.
9. David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2):91-110, 2004.
10. A. Hinneburg, D. A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. *KDD*, 58-65, 1998.
11. K. Mikolajczyk, C. Schmid. A performance evaluation of local descriptors. *CVPR*, 2:257-263, 2003.
12. Josef Sivic, Andrew Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. *ICCV*, 2:1470-1477, 2003.
13. George Forman. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research. 3:1289-1305, 2003.
14. Z.Q Bian, X.G. Zhang. Pattern Recognition. Tsinghua University Press, 2000.
15. V. Vapnik. Statistical Learning Theory. Wiley, 1998.

An Energy Minimization Process for Extracting Eye Feature Based on Deformable Template

Huachun Tan and Yu-Jin Zhang

Department of Electronic Engineering, Tsinghua University,
Beijing 100084, China
tanhc00@mails.tsinghua.edu.cn, Zhang-yj@tsinghua.edu.cn

Abstract. Eye feature extraction is of crucial importance for face recognition. Deformable template is an efficient model for this task. However, it usually suffers from the problem of local minima. To avoid local minima, in this paper, a new energy minimization process is proposed, which emphasizes on local properties of energy terms. The minimization process is divided into three steps. The iris is located firstly. Then the eye boundaries are adjusted. Finally, all energy terms are activated to tune the eye template. Each step needs not to be split to some sub-steps. Empirical comparison with other minimization processes shows the superiority of the proposed process in terms of both efficiency and accuracy.

1 Introduction

Eye feature extraction plays an important role in many applications, such as visual interpretation, recognition of human face [1], intelligent coding system and HCI (human-computer interface) [2, 3]. In the case of interpretation and recognition of human faces, most of attempts are made using geometrical features, where the relative positions and the shapes of the different features are measured. In HCI, the facial features, including the important part on face - eye, are extracted first, then these features are tracked to get the information of the facial expressions [3].

In eye feature extraction, the deformable template is an efficient model. Many methods [4-8] use deformable template to extract eye feature after the pioneer work of Yuille [9]. However, it always suffers from many problems, such as local minima, and low convergence speed. To improve the convergence speed, and to guarantee a good fit for avoiding local minima, minimization process has been extensively studied [4, 5, 7, 9] beside designing new energy function that can grasp the essence of eye feature. In this paper, we focus on the minimization process to improve the performance of eye feature extraction. The details of analyzing the problems about minimization process are described in section 2.

In this paper, a new minimization process is proposed to alleviate the problem of local minima. Considering the local properties of some energy terms, such as corner energy terms, the energy function of the deformable template is optimized in three epochs. The proposed method has been applied to real eye images.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 663–672, 2006.

The experiments show that the proposed method can balance the precise of eye feature localization and the time complexity.

The remainder of this paper is organized as follows. In Section 2, a review of some existing minimizing processes is provided. In Section 3, the geometric template and energy function used in our method is brief overviewed. The proposed minimization process is described in Section 4. The comparative experimental results for showing the superiority of the proposed method over some existing methods are presented in Section 5. Finally, the conclusions are given in Section 6.

2 Related Works

The existing minimization methods differ in both epochs and iteration methods. Summaries of three methods are given in Tables 1 to 3.

Table 1. Yuille *et al*'s method [9]

steps		Energy function	Parameters adjusted
Adjust iris	1	Valley	Iris location
	2	Valley, Intensity, Edge	Iris location and size
	3	Valley, Intensity, Edge	
Adjust eye Boundaries	4	Peak	Eye location and angle
	5	Peak, Intensity, Edge, Prior potential about eye boundaries	Eye location, size and angle
Finely tune	6-8	All energy term	All parameters

Table 2. Lam *et al*'s method [5]

steps		Energy function	Parameters adjusted
Adjust iris	1	Valley	Iris location and size
	2	Valley, Intensity, Edge	
Adjust eye Boundaries	3	Orientation,Boundary	Angle
	4	Edge, Intensity, Prior	Eye location and size
Finely tune	5	All energy term	All parameters

Some questions of these methods exist. First, is it necessary to divide each epoch into some sub-steps? In general, too many minimization steps would result in some parameters of the eye template being overly changed [4]. Some energy terms are conflicted, and only these conflicted terms reacted with the eye template would let the template deform correctly [4]. Second, whether the fewer epochs is the better? Some energy terms only work in small neighborhood. That is, only after the template is moved near the correct location, the energy

Table 3. Shan *et al*'s method [5]

steps		Energy function	Parameters adjusted
Adjust iris	1	Weighted edge map, Internal, Prior	Iris location and size
Adjust upper eyelid boundary	2	Weighted edge map, Internal, Prior	Upper eyelid boundary
Adjust lower eye boundary	3	Weighted edge map, Internal, Prior	Lower eyelid boundary

terms could interact with the input image to deform the template. For example, the corner energy term in [4] uses local information, which would interact with the input image only after the corners of the template near the right locations. If all energy terms are activated with the template in only one step, from our experiment results, the problem of local minima is serious and the convergence speed is slow in many cases.

Using too many epochs (or steps) in minimization process will result in parameters being overly changed, and using too few epochs is also harmful to the minimizing process. A minimization process with suitable epochs should be designed to minimize the energy function.

3 Geometric Model and Energy Function

3.1 Geometric Model

The eye template used in our system is parameterized as in Fig. 1 [8]. The circle is centered at X_c with radius r. The two half parabolas are centered at X_e, both with width b. The upper and lower parabolic curves have the heights a and c, respectively. These curves intersect at the four points P1, P2, P3, P4. Therefore, the eye template could be represented by $(Xc, Xe, r, a, b, c, \theta\ )$. All parameters are allowed to change.

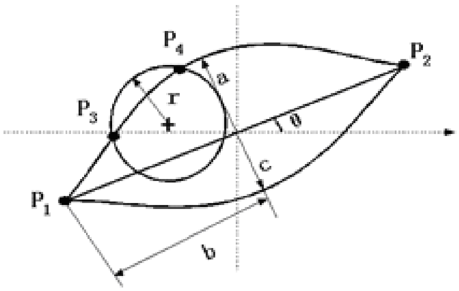


Fig. 1. The eye template in our system

3.2 Energy Function

The energy function used in our system is the same as [8] except using the corner strength function in SUSAN (Smallest Univalue Segment Assimilating Nucleus) [9] detector to define the corner energy term. The corner energy term is defined as

$$E_c = 1 - \frac{1}{n}\sum_{i=1}^{n} Corstr_i \ . \tag{1}$$

where $Corstr_i$ is the value of corner strength at the i-th corner, n is the possible number of corners in the eye template.

The corner strength in pixel (x, y), where x and y are the coordinate values in X and Y direction respectively, is defined as

$$Corstr_i = \begin{cases} g - n(x,y) \ ifn(x,y) < g \\ 0 \qquad\qquad\quad \text{otherwise} \end{cases} \tag{2}$$

where $n(x, y)$ is just the number of pixels in the USAN (Univalue Segment Assimilating Nucleus). The fixed threshold g (the "geometric threshold"), which is set to $n_{max}/2$ or even smaller to detect sharper corners, where n_{max} is the maximum value which n can take. To be consistent with other energy terms, which are normalized to (0, 1), the corner strength is also normalized. The details of SUSAN detector can refer to [9].

Then the energy function is defined in terms of the deformation of the template based on these fields. The details of other energy terms can refer to [4, 8].

4 Minimization Process

To avoid local minima, and to improve the convergence speed, improvement on minimization process for energy function is required. The new method is proposed based on the properties of energy function and our following observations. Fig.2 shows the valley, peak, edge fields and the corner strength field. It can be found that,

- The iris can be located rather well by only using the energy terms about the iris.
- Some energy terms, such as that on valley field, peak field and edge field, can interact with the template in big regions. These energy terms are the main forces that drag the template in a large region to the correct position and correct scale.
- Some energy terms, such as corner energy term, can only interact with the eye template in small regions. When the whole template is located in its neighborhood, the template can be refined by these energy terms. However, when the whole template is far from its location, the forces associated with these energy terms may drag the template to local minima.

Based on the observations mentioned above, and the discussion in Section 2, the following rules that should be considered in the minimization process are proposed:

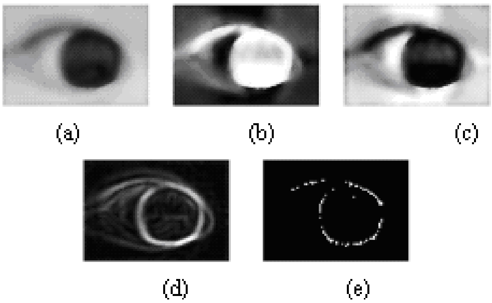


Fig. 2. The original image, valley, peak, edge fields and corner strength field of an eye image from (a) to (e)

- The iris should be located firstly. Then, the eye boundaries can be modified according to the location of iris.
- The energy terms that interact with the template in far ranges would be active prior to those in small ranges. The energy terms that interact with the template in far ranges can drag the template to its location, and avoid to local minima, while the energy terms that interact with the template in small ranges can adjust the template finely.
- Using as few as possible steps for minimization process to reduce the problem of over-changing parameters.

Then, a three-steps minimization process is proposed to optimize the energy function that would avoid the problem of local minima, that is,

Step 1. The iris is located in this step. The image intensity and edge forces for the circle are allowed to act on the template. In this step, (X_e, r) are updated and other parameters remain unchanged.

Step 2. The eyelids are adjusted to the correct location. The eyelids can be rotated, translated and resized by using the exterior forces, i.e., the forces except the interior force that only adjust the shape of the eye template. In this stage, the corner strength image force would react with the input image, (X_c, a, b, c, θ) are tuned in this stage and the parameters about the iris, (X_e, r), remain unchanged.

Step 3. In this step, all parameters are finely tuned by considering all the energy terms.

Table 4. Proposed minimization process

steps		Energy function	Parameters adjusted
Locating iris	1	Valley, Intensity, Edge	The parameters about iris
Adjust eye boundaries	2	Edge, Intensity, Peak, Corner, Prior	The parameters about eye boundaries
Finely tune	3	All energy term	All parameters

In the processing of minimization, each step needs not to be split into some sub-steps. This proposed method is summarized in Table 4.

5 Experimental Results

5.1 Measurement

In order to evaluate the performances of locating eye features quantitatively, the statistic errors of corner location is used in this paper because the location of eye corners can reflect the structure of eye feature effectively. The error of corner location is defined as the average Euclidean distance between the ground truth and the detected location, i.e.

$$Errcor_j = \frac{1}{N}\sum_{i=1}^{N} \left\| X_{truth_i,j} - X_{extracted_i,j} \right\| . \tag{3}$$

where$X_{truth_i,j}$ represents the location of ground truth of j-th eye corner in the i-th eye image, and $X_{extracted_i,j}$ represents the location of j-th eye corner extracted from the i-th eye image. N is the number of examined eye images. $Errcor_j$ means the error value of j-th corner.

5.2 Experimental Results

The proposed algorithm has been implemented using Matlab and applied to real images. In the experiments, totally 120 eye images, in which 110 images selected from the Pitt-CMU Facial Expression AU Coded Database [11] and 10 images downloaded from Internet, are used. The typical image size is 101x56, the radius of the circle for computing corner strength is 3. The initial parameters of eye feature are determined manually. Firstly, six points from the original image are selected manually. Then the ground truth of the eye template is calculated. Finally, the ground truth is displaced by a random variable as initial parameter to simulate the real situations. Based on the initial parameters, the eye feature is extracted through minimization process. Fig. 3 shows a sequence of eye templates at the end of each step. In each step, the parameters obtained from the previous step are taken as the initial parameters for the current step.

The performances of extracting eye feature using proposed minimization process are compared with those of other minimization processes while using same energy function proposed in [8]. Both one-step minimization process that likes Xie *et al*'s method [4], and 5-step minimization that likes Lam *et al*'s method [5] are taken into account here. To compare the behavior, these methods are all use the same pre-process and initial parameters. The statistic results are reported in Table 5. The first two columns show the distances of inner eye corner and outer eye corner extracted from the ground truth respectively (unit is pixel). The third column is the mean of the value in the first two columns. Some results are shown in Fig. 4. For convenience, we only give the final extraction results.

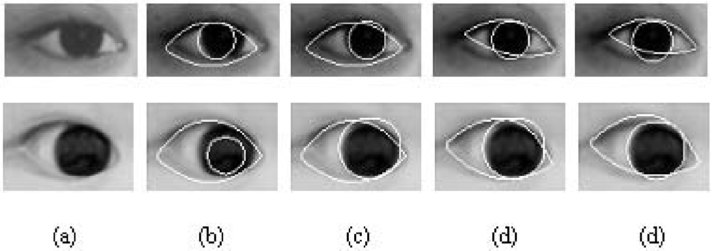


Fig. 3. Eye templates at the end of each step. (a) Original image (b) Initial template (c) Result of step 1 (d) Result of step 2 (e) Result of step 3.

Table 5. Average error of corner of some minimization process (unit is pixel)

	Inner corner	Outer corner	Mean
1-step	2.3	2.0	2.2
5-step	3.8	3.7	3.8
Proposed	1.5	2.0	1.8

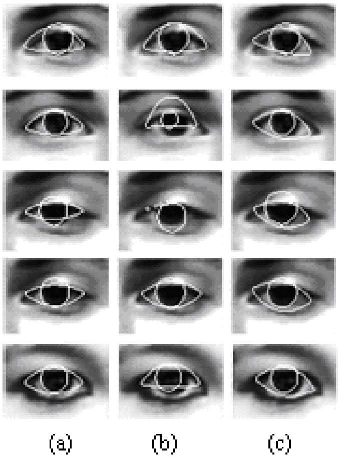


Fig. 4. Results of comparison using proposed energy function. (a) 3-steps (b) one-step (c) 5-steps.

From the experiments, the proposed method using 3-steps minimization process gets the best results.

From Fig. 4, it could be found if too many steps were applied, some parameters would be over changed. For example, in Fig. 4(c), the eye boundaries always tilt in the same direction, which maybe caused by overly changed angle parameter. If only one step is used to minimize the energy function, the eye template also fall into local minima due to the disturbance of the energy terms that only interact

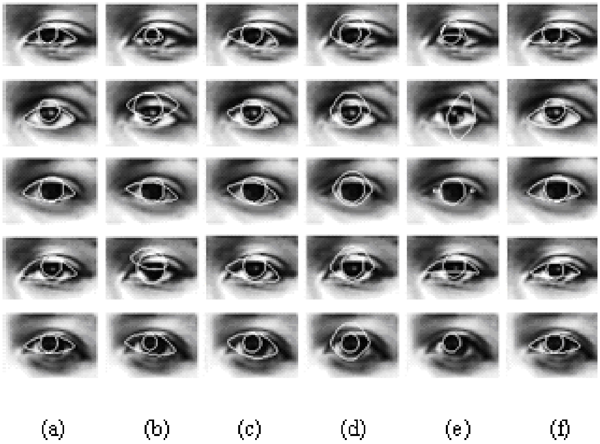


Fig. 5. Results of comparison (a) Proposed (b) Xie *et al* [4] (c) Lam *et al* [5] (d) Shan *et al* [7] (e) One step (f) Tan *et al* [8]

Table 6. Average error of corner of some methods(unit is pixel)

	Inner corner	Outer corner	Mean
Xie *et al* [4]	6.2	5.8	6.0
Lam *et al* [5]	4.4	4.4	4.4
Shan *et al* [7]	5.1	5.2	5.1
Tan *et al* [8]	3.3	3.8	3.5
Proposed method	1.5	2.0	1.8

with eye template in small range, such as corner energy terms. However, the proposed method can extract the eye template more robustly.

We also compared our proposed method with that of other eye feature extraction methods using the same eye images. The compared method are Xie *et al* [4], Lam *et al* [5], Shan *et al* [7], Tan *et al* [8] and proposed method. Fig. 5 shows some examples of the comparisons. The statistic value about corner location from the ground truth is given in Table 6.

Xie's method also fails in extracting eye boundaries in many cases. The reason is also the local property of corner energy term. The method proposed by Lam *et al* [5] could extract the eye boundaries but fail to extract the right corners. This may be caused by the energy function. Because only edge and prior potential are used by Shan's method, and not having a finely tuned process, it is sensitive to initial parameters and always falls into local minima. Both Tan's method [8] and proposed method use the same minimization process, their performances outperform others. And the proposed method is the best because the corner energy function using SUSAN corner strength function can represent the properties of corner more efficiently.

Table 7. Comparison of process time (unit is second)

Method	Xie *et al* [4]	Lam *et al* [5]	Shan *et al* [7]	One step	Tan *et al* [8]	Proposed
Time	57.8	66.6	5.1	52.6	54.1	33.8

In addition, we have compared the convergence time of these methods. Table 7 shows the mean time used in optimization process. The speed of Shan's method is far faster than others are. The reason is the simplification of energy function. However, the accuracy of Shan's method is poor. In the remaining methods, our method is faster than others are. This is because only fewer parameters require modifications in the first two steps, and the parameters change only within a very small region in the third step in the proposed method.

6 Conclusion

In this paper, a three-steps minimization process for energy functions has been proposed to alleviate the problem of local minima by considering the local properties of energy terms. Experimental results show that our minimization method works well, and justify its superiorities over the existing methods.

Comparing with previous minimization process using the same energy function, such as 1-step and 5-steps minimization process, the proposed method improved the accuracy of locating eye corners about 22% and 111%, respectively. Comparing with other eye feature methods using different energy function and minimization process, such as the methods proposed by Xie *et al* [4], Lam *et al* [5], Shan *et al* [7] and Tan *et al* [8], the mean error of proposed method is reduced to 30%, 41%, 35% and 51%, respectively, while the speed was faster than Xie *et al*'s [4] and Lam *et al*'s [5] methods.

Currently, there are still a number of coefficients in the different energy terms that should be determined before minimization. This is our future work.

Acknowledgment

This work has been supported by Grant RFDP-20020003011.

References

1. Zhao,W. Chellappa, R. Phillips, P.J. and Rosenfeld, A. Face Recognition: A Literature Survey, *ACM Computing Surveys*, 35(4): 399-458, 2003.
2. Fasel, B. and Luettin, J. Automatic Facial Expression Analysis: A Survey, *Pattern Recognition*, 36(1): 259-275, 2003
3. Tian, Y. Kanade, T. and Cohn, J. Recognizing Action Units for Facial Expression Analysis, *IEEE Transaction on Pattern Analysis and Machine Intelligence*,23(2): 97-115, 2001

4. Xie, X. Sudhakar, R. and Zhuang, H. On Improving Eye Feature Extraction Using Deformable Templates, *Pattern Recognition*, 27(6): 791-799, 1994.
5. Lam, K. and Yan, H. Locating and Extracting the Eye in Human Face Images", *Pattern Recognition*, 29(5): 771-779, 1996.
6. Deng, J. and Lai, F. Region-based Template Deformation and Masking for Eye-feature Extraction and Description," *Pattern Recognition*, 29(3): 403-419, 1997
7. Shan, S. Gao, W. and Chen, X. Facial Feature Extraction Based on Facial Texture Distribution and Deformable Template, *Journal of Software*, 12(4): 570-577, 2001
8. Tan, H. Zhang Y. and Li R. Robust Eye Extraction Using Deformable Template and Feature Tracking Ability, *ICICS-PCM*, 3: 1747-1751, 2003.
9. Yuille, A. L. Cohen D. S. and Hallinan P. W. Feature Extraction from Faces Using Deformable Templates, *CVPR*, 104-109, 1989.
10. Smith, S. and Brady, J. SUSAN - A New Approach to Low Level Image Processing, *International Journal of Computer Vision*, 23: 45-78, 1997.
11. Kanade, T. Cohn J.F. and Tian Y. Comprehensive Database for Facial Expression Analysis, *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 46-53, 2000.

Image Feature Detection as Robust Model Fitting

Dengfeng Chai[1,2] and Qunsheng Peng[1]

[1] State Key Lab of CAD&CG, Zhejiang University, Hangzhoz 310027, China
{chaidf, peng}@cad.zju.edu.cn
[2] Institute of Space and Information Techniques, Zhejiang University,
Hangzhou 310027, China

Abstract. In this paper, we describe image feature as parameterized model and formulate feature detection as robust model fitting problem. It can detect global feature easily without parameter transformation, which is needed by Hough Transform methods. We adopt RANSAC paradigm to solve the problem. It is immune to outliers and can deal with image contains multiple features and noisy pixels. In the voting stage of RANSAC, in contrast with previous methods which need distance computation and comparison, we apply Bresenham algorithm to generate pixels in the inlier region of the feature and use the foreground pixels in this region to vote the potential feature. It greatly improves the efficiency and can detect spatially-linked features easily. Experimental results with both synthetic and real images are reported.

1 Introduction

Image feature detection is an important topic in computer vision. Given a gray or color image, edge detection can be applied to detect edges and output an edge image which is a binary image of edge (foreground) pixels and non-edge (background) pixels. Detecting features in this binary image is a difficult problem and is the focus of this paper. The methods proposed up to date are categorized into segment grouping based methods [1, 2] and Hough Transform methods (HT) [3, 4, 5, 6, 7].

Segment grouping based methods consist of two stages: linking foreground pixels into segment elements and grouping these elements into global features. Since the grouping criteria are locally optimal, the performance of detecting global features is poor.

In contrast with segment grouping based methods, HT methods map foreground pixels into parameter space and detect features in parameter space. They consist of voting and searching stages, i.e. mapping foreground pixels into accumulators in parameter space and detecting maximal value in accumulators. Because pixels belong to one feature are mapped to one accumulator, they can detect global features successfully at the cost of great storage for accumulators in parameter space and computation time for voting and searching process. Besides, the spatial relationship of foreground pixels is lost in the voting stage.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 673–682, 2006.

To save computation time, Probabilistic HT (PHT) [5] selects a pre-selected proportion of the foreground pixels in original image for voting. The time saved depends on the ratio of selected pixels with respect to all foreground pixels. Too small proportion frequently leads to incorrect detection results. To select a proper proportion, a priori knowledge about the image is needed. Progressive PHT (PPHT) [7] requests no a priori knowledge, it selects pixels randomly for voting, removes the foreground pixels from image and un-votes accumulator once a highest peak and corresponding line segment is detected. To alleviate the extra storage requirement, Random HT (RHT) [6] adopts many to one mapping and list structure techniques. The computation time is also saved by these techniques.

Chen and Chung have modified RANSAC and developed Random Line Detection (RLD) [8] and Random Circle Detection (RCD) [9] algorithms. They select three or four foreground pixels respectively to define a line or circle and use the left pixels to vote the defined feature. They can detect features with no need of parameter transformation. But the algorithms are inefficient because of explicit distance computation involved in the voting stage. Besides, the spatial relationship between foreground pixels is not well utilized. Zhang have investigated different parameter estimation techniques and presented a tutorial focusing on conic fitting [10].

Motivated by RLD and RCD, we formulate image feature detection as robust model fitting problem in this paper: treat foreground pixels as data points, use parameterized model to describe the image features (such as lines and circles), and treat feature detection as model fitting. Since the global information is implicated in the parameterized model, it can detect global features easily. We adopt RANSAC [11] to solve to the fitting problem, RANSAC is a robust method and is immune to outliers in the original data points, therefore it can detect feature from image contains multiple features and noise pixels. In the voting stage of RANSAC, instead of checking all foreground pixels to vote the feature, we adopt Bresenham algorithm [12] to generate pixels within *inlier region* of the feature and use foreground pixels in this region to vote the feature. This avoids explicit distance computation and improves efficiency greatly. Besides, it detects features directly in image space without involving parameter transformation, therefore needs no extra time and storage requirement. The successive pixels generated by the Bresenham algorithm are spatially neighboring, this property is easily utilized to detect spatially-linked features.

We formulate image feature detection as robust model fitting problem in section 2, propose the solution in section 3, and then present the detection algorithm in section 4. After that, we show experiment results in section 5 and draw conclusion in section 6.

2 Problem Formulation

2.1 Feature Representation

As shown in Fig 1, there are many foreground pixels in the image, some pixels form line l_1, l_2 and l_3 , some form ellipse e, some form circle c and some form

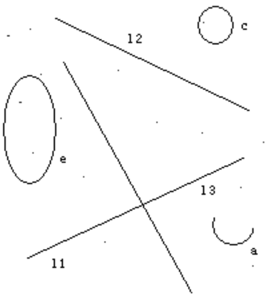


Fig. 1. Image feature detection as robust model fitting

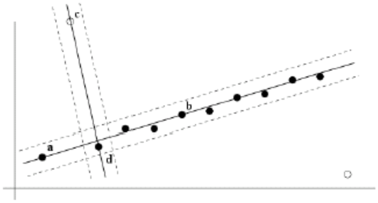


Fig. 2. RANSAC for line fitting: the dotted lines indicate the threshold distance

circle arc a, and others are just noise pixels. The lines, circle, ellipse etc. are image features to be detected. All pixels on one feature satisfy Eq.(1):

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{1}$$

where a, b, c, d, e, f are free coefficients, so, the features can be described as Eq.(1) with a, b, c, d, e, f being specified.

Eq.(1) is a conic equation describes general curves including circle, ellipse and etc. And these curves are just specific conic with their coefficients meet some constraints. For example, if $a = c, b = 0$, then the conic degrades to be a circle, if $a = b = c = 0$, then the conic degrades to be a straight line. In this paper, we represent image features as equations like that of Eq.(1) and call it as *model representation*. We deal only with image features that can be described by the parameter equation. The model has free coefficients a, b, c, d, e, f and their specified values defines an image feature.

2.2 Image Feature Detection as Robust Model Fitting

Let us assume at first that there is only one line l_1 in the image shown in Fig.(1) and we want to detect it. As shown in subsection 2.1, l_1 can be represented by Eq.(1) with $a = b = c = 0$ and d, e, f being specified. What left to do is to specify the free coefficients d, e, f, it is a well-known *model fitting* problem: fit a model to the pixels so that the distance of the pixels deviated from the model is minimized.

But there are l_2,l_3, etc. together with many noise pixels in the image, fitting a model to all the foreground pixels is meaningless and can not detect the features at all. It is necessary to distinguish pixels which belong to l_1 from other pixels first. Once this is done, the model fitting methods can be applied to fit a model to the distinguished pixels. From the point of view of model fitting, all pixels on line l_1 are inliers to l_1 while other pixels are outliers. The model fitting method must be robust enough to deal with cases there are outliers in the original pixels. It is the nature of *robust model fitting* problem [13].

Since there are many features in the image, it is necessary to carry out the model fitting method repeatedly until all the features are successfully detected.

3 Solution to the Feature Detection

In the previous section, we formulate the image feature detection as robust model fitting problem. There are lots of methods designed to solve this problem [10, 11]. RANSAC can cope with a large proportion of (more than 50%) outliers. As shown in 2.2, since pixels in l_2, l_3 etc. are outliers with respect to l_1, there are usually more than 50% outliers in the original data to be fitted. We adopt the RANSAC algorithm in this paper to solve the model fitting problem.

3.1 RANdom SAmple Consensus

RANSAC does trial repeatedly to find the model. Each trial consists of sampling and voting stages. In the sampling stage, it randomly selects a minimal subset of the original data points and instantiates a model from the subset. In the voting stage, it determines the consensus set of the determined model by distinguishing the set of data points within a distance threshold of the determined model from other points. The termination condition is either a model is found successfully or the number of trials reaches a preset threshold. The algorithm is presented as follow:

1. Set $C_{sample} = 0$, while $T_t > C_{sample}$ do 2-5:
2. **Sampling stage:** Randomly select a sample of s points from original data points and instantiate a model from the selected points,
3. **Voting stage:** Determine the consensus set (set of inlier points) which contains points within a distance threshold T_d of the model,
4. If the size of the consensus set is greater than a preset threshold T_c, report the model and terminate,
5. Let $C_{sample} = C_{sample} + 1$,
6. The largest consensus set is selected as inliers and corresponding model is selected as the final model.

where C_{sample} is the counter for trial number, s is the minimal number of points needed to determine a model. T_d depends on the required fitting precision, T_c is a function of number of inliers and T_t is specified by:

$$T_t = \lg(1-p)/\lg(1-\epsilon^s) \tag{2}$$

where p is the probability that at least one random sample is free of outliers, it is always chosen as 0.99, ϵ is the proportion of inliers.

Fig. 2 illustrates how RANSAC fit a line to the data points. It randomly selects 2 points to define a line, points between the two dashed lines parallel with the defined line are within a distance threshold to the line and form the consensus set. As shown, the size of consensus set of line (a, b) is 10 while that of line (c, d) is 2, so, RANSAC selects line (a, b) as the fitting result at last.

3.2 Sample Minimal Set

It might be thought that it would be preferred to use more than minimal subset to instantiate a model as RLD [8] and RCD [9] do, because a better estimate of the model would be obtained from them, and the measured support would reflect the true support more accurately. However, this possible advantage in measuring support is generally outweighed by the severe increase in computational cost incurred by the increase in the number of trial.

Because there are often lots of pixels in the image, it is computationally infeasible to try every possible sample in the sampling stage. In fact, it is unnecessary to enumerate all the possible samples exhaustively. Instead the necessary number of samples T_t is chosen sufficiently high to ensure that at least one of the random sample of s points is free from outliers with a probability of p. Eq.(2) shows the relationship between T_t and p, ϵ, s. Given an image, the ϵ is constant with respect to the feature to be detected, the p is also constant in the detection process (it is always chosen as 0.99), so, T_t increases exponentially with s. Tab. 1 shows an example of T_t for given s and e. As shown, the necessary number of trials increases dramatically with s increasing, therefore the computation cost is increased severely.

Based on these observation, we follow the minimal set principle, i.e. select minimal number of points needed to determine the model to be found.

3.3 Instantiate Model from Minimal Set of Points

The minimal number of points needed to instantiate a model is equal to the number of free coefficients in the model representation of the feature to be detected. For example, it is 2 for straight line, 3 for circle and 4 for ellipse.

Given a minimal set of points, the model is instantiated by solving the unknown coefficients in the equations for the model. Suppose that (x_1,y_1),...,(x_5, y_5) is selected as minimal set, then (x_i, y_i) is on the conic and we have:

$$A_i X = 0 \tag{3}$$

with

$$A_i = \begin{bmatrix} x_i^2 \; x_i y_i \; y_i^2 \; x_i \; y_i \; 1 \end{bmatrix} \tag{4}$$

$$X = \begin{bmatrix} a \; b \; c \; d \; e \; f \end{bmatrix}^T \tag{5}$$

Stacking equations from each point $(x_i, y_i), i = 1, ..., 5$ in to one set of equations, we get:

$$AX = 0 \tag{6}$$

$$A = \begin{bmatrix} A_1^T \cdots A_5^T \end{bmatrix}^T \tag{7}$$

the unknown $X = (a, b, c, d, e, f)^T$ is a homogeneous vector and has only 5 degrees of freedom, so, it can be solved from the 5 equations in Eq.(6). Since Eq.(6) is a set of homogeneous equations and the obvious solution $X = 0$ is meaningless, it can be solved by putting an additional conditional on the norm of the unknown vector, e.g. $\|X\| = 1$. Instead, we turn Eq.(6) into a inhomogeneous set of equations by imposing a condition $X_i = 1$ for one unknown and some other conditions on the other unknowns. For example, in the case of a circle, a in Eq.(1) is sure to be nonzero, therefore the additional condition $X_1 = a = 1$ can be imposed. Further, $b = 0$ and $c = a$ can also be imposed for a circle. The number of free unknowns is left to be only 3. Based on these conditions, Eq.(8) can be derived from Eq.(6), it have 3 linear equations and 3 unknowns, the unknowns can be solved easily.

$$\begin{bmatrix} x_1\, y_1\, 1 \\ x_2\, y_2\, 1 \\ x_3\, y_3\, 1 \end{bmatrix} \begin{bmatrix} d \\ e \\ f \end{bmatrix} = \begin{bmatrix} -(x_1^2 + y_1^2) \\ -(x_2^2 + y_2^2) \\ -(x_3^2 + y_3^2) \end{bmatrix} . \tag{8}$$

3.4 Determine the Number of Samples Adaptively

The proportion of inliers ϵ is often unknown because we do not have statistics of foreground pixels and features in advance. Further more, ϵ is different with respective to different features and is varying while the detection process proceeds. Therefore, T_t can not be determined in advance.

We apply an adaptive strategy to solve this problem, i.e. determine ϵ and T_t adaptively while detection proceeds. It records the maximal value of ϵ and use it to determine the necessary number of trials. The adaptive algorithm is as follows:

1. Let $T_t = \infty$, $\epsilon = 0$, $\epsilon_{max} = 0$ and set $C_{sample} = 0$.
2. While $T_t > C_{sample}$ do 3-7:
3. Sampling stage,
4. Voting stage,
5. Let $\epsilon = N_{inlier}/N_{total}$, $\epsilon_{max} = max(\epsilon_{max}, \epsilon)$,
6. Compute T_t from ϵ_{max} using Eq.(2),
7. Let $C_{sample} = C_{sample} + 1$.

where ϵ_{max} records the maximal value of ϵ, N_{inlier} is the number of inlier points found in each trial while as N_{total} is number of all points.

3.5 Voting Without Explicit Distance Computation

As shown in subsection 3.1, in the voting stage of RANSAC, it needs to determine the consensus set and this needs distinguishing points within a distance threshold T_d of the model from other points. Obviously, it needs distance computation and comparison which consume much time, and this is what previous method really do. In this section, we will show how the distance computation can be avoided and present a new voting method without involving explicit distance computation.

In fact, the voting stage needs only counting points within a region we called *inlier region* which centers at the model and dilates from it with diameter T_d.

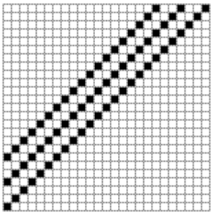


Fig. 3. Inlier region of a model: the center line indicates a model, the region between the up and below line indicates the inlier region and it contains only limited pixels

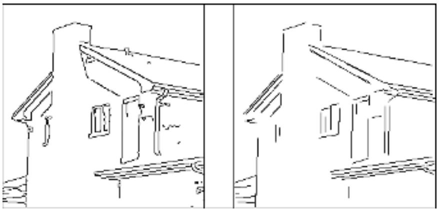


Fig. 4. Left image is an edge image of a house, and right image shows the detected line segments by our method

To do this, previous methods check all points by computing their distance from the model and count the ones whose distance smaller than T_d. But as shown in Fig.3, images contain only discrete pixels, and there are limited pixels in the inlier region. All the pixels in the inlier region are either foreground or background pixel, and only foreground pixels in the inlier region vote the model. Therefore, the alternative is checking all the pixels in the inlier region, if the pixel is foreground, it votes the model.

Now, let's assume that the model to be fitted is straight line. As shown in Fig.3, the inlier region of the line model is the region between two lines deviate from the model with a distance T_d. This region is equivalent to a line with width $2T_d$ centered at the model. This is also true for other models, therefore we have:

The inlier region of a model M is equivalent to a model M_e with width $2T_d$ centered at M.

Generating a line or curve with a width is a standard rasteration problem in computer graphics. Bresenham algorithm [12] is a widely used algorithm for rasteration, it can be implemented with only integer calculations and is fast. There are Bresenham algorithms [14] designed to generate straight line, circle and ellipse etc.

In this way, the distance computation and comparison is avoided, and this saves much computation time as will be shown in section 5. Furthermore, there is another advantage for applying rasteration method as an alterative to distance computation as shown in next subsection.

3.6 Explore Spatial Information in the Voting Stage

In fact, series of pixels are generated pixel by pixel in the rasteration methods. As shown in Fig.3, for example, Bresenham algorithm generate pixels from left to right, the successive pixels are spatially connected. Apparently, it is easy to record the consecutive foreground pixels and consecutive background pixels. In this way, we can detect spatially linked segments of straight line or curve. To account for noisy pixels in the original data and errors in edge detection, it should allow small gaps between segments. We set a threshold T_g for the gap between segments, segments with gap smaller than T_g are merged to be one segment. Therefore, it does not need post-processing which is needed by RLD, RCD and most Hough Transform method.

4 Robust Model Fitting Based Feature Detection Algorithm

In this section, we present the proposed feature detection algorithm as follows:

1. Collect all foreground pixels in image I into set S and let N_{total} be the size of S,
2. Let $T_t = \infty$, $\epsilon = 0$, $\epsilon_{max} = 0$, $C_{sample} = 0$,
3. While $T_t > C_{sample}$ do 4-10:
4. Randomly select s points from S,
5. Determine a model M from the selected s points using method described in subsection3.3,
6. Apply Bresenham algorithm to generate pixels inside the inlier region of model M,
7. Count the number of spatially-linked foreground pixels of the generated pixels as N_{inlier},
8. Let $\epsilon = N_{inlier}/N_{total}$, $\epsilon_{max} = max(\epsilon_{max}, \epsilon)$,
9. Compute T_t from ϵ_{max} using Eq.(2),
10. Let $C_{sample} = C_{sample} + 1$,
11. If $N_{inlier} > T_{inlier}$ do 12-15:
12. Report the detected feature,
13. Remove the pixels on the detected feature from image I and corresponding data points from set S,
14. Let $N_{total} = N_{total} - N_{inlier}$,
15. go back to 2
16. Terminate.

where, T_{inlier} is a preset threshold for the minimal number of pixels one feature should have.

5 Experiments and Comparison

Based on section 4, we develop a Robust Model Fitting Based Line Detection method (RMFBLD) and apply it to both synthetic and real images to test its correctness and efficiency. Size of synthetic images is 256 × 256. The number of line segments in one image is used to control complexity of image. It ranges from 10 to 50 using 10 as step. Noise level is characterized by number of noise pixels. It ranges from 0 to 500 by a step of 50. For every level, 32 images are synthesized using different random seed. PPHT, RLD and RMFBLD are applied to detect line segments in these images. The total number of detected line segments and time used are shown in Tab. 2. As shown, RMFBLD is the most efficient method. Fig.5 shows one example of the results, it has 30 line segments and 300 noisy

Table 1. The necessary number of samples T_t for a given s and e

s	e				
	90%	80%	70%	60%	50%
2	3	5	7	11	17
3	4	7	11	19	35
4	5	9	17	34	72
5	6	12	26	57	146
6	7	16	37	97	293

Table 2. Comparison of RMFBLD with RLD and PPHT Method

Method	Detected line segments	Time (second)	Lines per second
RLD	3494	52,793	66.2
PPHT	18127	503,222	36.0
RMFBLD	18769	127,357	147.4

pixels, the original and detected line segments using RMFBLD, PPHT and RLD are shown from left to right, as shown, RMFBLD can detect features from image contains multiple features. As can be seen in both Tab. 2 and Fig.5, TRMFBLD and PPHT detect approximately the same number of lines, but RLD detects less lines.

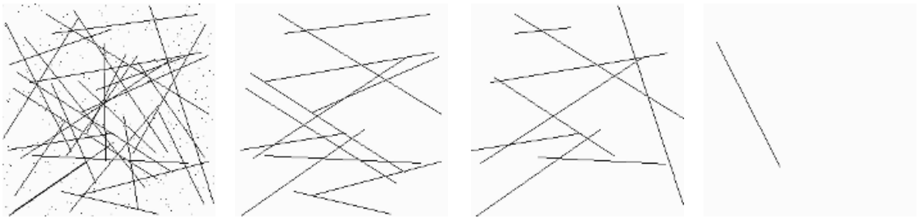


Fig. 5. Result example

Fig.4 shows another example, left figure is an edge image of an image of a house, and right one shows detected line segments by RMFBLD. It can be seen that global line features are successfully detected.

6 Conclusion

We formulate feature detection as robust model fitting problem. First, we use parameterized model to describe image features, and treat feature detection as model fitting problem. The global information is implicated in the parameterized model, global features can be easily detected without involving parameter transformation. Second, we adopt RANSAC as a solution to the model fitting problem. Because RANSAC is immune to outliers, the proposed method can deal with images contains multiple features and noisy pixels. Third, we develop a novel voting method for RANSAC, it avoids explicit distance computation by generating inlier pixels and checking if they are foreground. Besides the efficiency improvement, it provide a good chance to detect spatially connected feature.

Apart from presenting the framework of robust model fitting based image feature detection, we also develop Robust Model Fitting Based Line Detection method for line detection at present. We plan to develop another method for detecting other features, such as circle and ellipse in the near future.

Acknowledgments

It is supported by the National Natural Science Foundation of China under Grant Nos.51476070101JW0409; the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312101. The authors would like to thank the anonymous reviewers for very useful comments.

References

1. Boldt, M., Weiss, R., Riseman, E.: Token-based extraction of straight lines. IEEE Trans. System, Man, Cybernet **19** (1989) 1581–1594
2. Nacken, P.: A metric for line segments. IEEE TPAMI **15** (1993) 1312–1318
3. Illingworth, J., Kittler, J.: Survey: Survey of the hough transforms. Computer Vision, Graphics, and Image Processing **44** (1988) 87–116
4. Leavers, V.: Survey: Which hough transform. Computer Vision, Graphics, and Image Processing: Image Understanding **58** (1993) 250–264
5. Kiryati, N., Eldar, Y., Bruckstein, A.: A probabilistic hough transform. Pattern Recognition **24** (1991) 303–316
6. Xu, L., Oja, E., Kultanan, P.: A new curve detection method: Randomized hough transforms (rht). Pattern Recognition Letters **11** (1990) 331–338
7. Matas, J., Galambos, C., Kittler, J.: Progressive probabilistic hough transform. In: Proc. British Machine Vision Conference. (1998)
8. Chen, T., Chung, K.: A new randomized algorithm for detecting lines. Real-Time Imaging **7** (2001) 473–481
9. Chen, T., Chung, K.: An efficient randomized algorithm for detecting circles. Computer Vision and Image Understanding **83** (2001) 172–191
10. Zhang, Z.: Parameter estimation techniques: A tutorial with application to conic fitting. Image and Vision Computing **15** (1997) 59–76
11. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. ACM Commun. Assoc. Comp. Mach. **24** (1981) 381–395
12. Bresenham, J.: Algorithm for computer control of a digital plotter. IBM System Journal **4** (1965) 25–30
13. Rousseeuw, P.J.: Robust Regression and Outlier Detection. Wiley, New York (1987)
14. Rogers, D.F.: Procedural Elements for Computer Graphics. McGraw-Hill (1998)

Extraction of Salient Contours Via Excitatory-Inhibitory Interactions in the Visual Cortex

Qiling Tang[1], Nong Sang[1], and Tianxu Zhang[2]

[1] Institute for Pattern Recognition and Artificial Intelligence,
Huazhong University of Science and Technology, Wuhan 430074, PR China
tqlinn@sohu.com, nsang@hust.edu.cn
[2] Key Laboratory of Ministry of Education for Image Processing and Intelligent Control,
Huazhong University of Science and Technology, Wuhan 430074, PR China
txzhang@hust.edu.cn

Abstract. In this paper we mimic a biological visual strategy to extract salient contours from complex scenes. Psychophysical and physiological studies show that the response to the stimulus within the receptive field is affected by the presence of surrounding stimuli— the response is suppressed significantly by similarly oriented stimuli in the surround while this suppression is converted to strong facilitation with the addition of collinear stimuli in the surround. According to this property of visual perception, we enhance salient contours and at the same time reduce the interference of the extraneous elements. Our results show the feasibility of the proposed method.

1 Introduction

It has long been a puzzle that how to automatically extract contours from complex scenes. This is because it requires distinguishing contours from non-contour edges, and grouping local elements into meaning global features. The human visual system presents an outstanding ability to contour processing, thus understanding the visual mechanisms can provide a biologically motivated scheme for contour extraction in computer vision.

A number of studies have shown that the stimuli outside classical receptive field (RF) exert a significant influence over the activities of cells in the primary visual cortex. Knierim and van Essen [1] observed experimentally that the response to stimulus in the RF is suppressed significantly by similarly oriented stimuli in the surround, i.e., iso-orientation suppression, and the suppression is reduced when the orientations of the surround stimuli are random or different from the stimulus in the RF. However, if the surround stimuli are aligned with the optimal stimulus inside the RF to form a smooth contour, then suppression becomes facilitation [2],[3]. Whether the response to stimulus presented within the receptive field can be facilitated or suppressed by other stimuli falling outside the receptive field depends on the relative orientation of stimuli inside and outside the receptive field [4]. Bonneh and Sagi [5] found that detectability depends on stimulus geometry and is constrained by collinearity and proximity spatial relationships, and therefore a coherent' configuration is more easily detected than a 'non-coherent' one.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 683 – 691, 2006.

In this paper, we design a model for contour extraction according to the above description of visual cortex. Compared to our previous work [6], the model is more reasonable. The improvement is as follows: I) for excitatory interactions, previous short-range connections are taken place by long-range horizontal ones, an important aspect of which is their ability to provide more excitatory inputs to their postsynaptic cells, which contribute to link local elements forming collinear alignment via more local information; II) for inhibitory interactions, we consider bow-tie-shaped inhibitory regions located in the side of the RF rather than ring-formed regions surrounding the RF, and the separation between inhibitory and excitatory regions void the suppression among collinear elements.

2 Algorithm Implementation

A complex cell in V1 can be regard as a local oriented energy operator. Gabor energy that is defined as the square root of the sum of responses of odd-even pairs of Gabor filters is used to represent the initial responses of V1cells. V1cells dynamically tune their responses according to contextual interactions, and then make local elements group into global features.

2.1 Collinear Excitation

While stimuli in the surround are located along the axis of the RF and share similar orientation tuning, they will enhance the responses of V1 cells. The two enhanced conditions are named as axial specificity and modular specificity by Shouval et al. [7]. Thus facilitation depends on the precise spatial alignment of the RF and surround, and the facilitatory effect decreases with the distance between the surround and the RF.

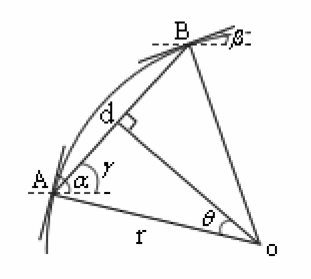


Fig. 1. The geometrical relationship based on the co-circularity rule

We combine the co-circular constraint proposed by Parent and Zucker [8] with the visual preference for low curvature to define a local grouping function of contour integration. Fig.1 illustrates the geometrical relationship based on the co-circularity constrain. A and B denote the cells inside and outside the RF respectively, and their spatial coordinates are (x, y) and (x', y') respectively. The edge elements belong to the same physical contour satisfy the co-circularity constraint to a large extent [9]. If the preferred orientation of A is α, then in term of the co-circular relationship, the orientation of B should satisfy the following,

$$\beta = \begin{cases} 2\gamma - \alpha + \pi & if \quad 2\gamma - \alpha < 0 \\ 2\gamma - \alpha & if \quad 0 \le 2\gamma - \alpha < \pi \\ 2\gamma - \alpha - \pi & if \quad \pi \le 2\gamma - \alpha \end{cases} \tag{1}$$

where γ is the angle of the line connecting the two cells A and B, and assume that $0 \le \alpha < \pi$ and $0 \le \gamma < \pi$. When the preferred orientation of B is β, B and A can form a co-circular smooth contour. The closer B's dominant orientation (i.e., corresponding to the orientation of the strongest response over all the orientations of B) approaches β, the stronger its response at the orientation β (relative to other orientations), and the greater the element with A produces collinear excitatory effect, otherwise, the weaker. Thus we can consider B's response at the orientation β only, and the amplitude of the response reflects the degree of both the elements satisfying the co-circularity.

Curvature is an important factor determining natural contour detectability. The dynamic process of contour integration varies with the curvature of the contour, and in general, visual sensitivity to contours increases with the length and straightness of the path [10]. The curvature k of the co-circular elements is given by,

$$k = \frac{1}{r} = \begin{cases} \frac{2}{d}\sin\left|\frac{\beta-\alpha}{2}\right| & 0 \le 2\gamma - \alpha < \pi \\ \frac{2}{d}\cos\left|\frac{\beta-\alpha}{2}\right| & 2\gamma - \alpha < 0 or 2\gamma - \alpha \ge \pi \end{cases} \tag{2}$$

where $d = \sqrt{(x'-x)^2 + (y'-y)^2}$.

In addition, excitatory regions should be disposed along the axis of the cell according to the axial specificity. Consequently, the deviation of the connecting line AB from the orientation axis of A is limited, i.e., $|\gamma - \alpha| \le \varphi$, φ is the upper bound of the angular deviation.

A curvature-based weighting function which reflects visual preference for the path with low curvature is expressed as follows,

$$W_c(x', y', \beta; x, y, \alpha) = \exp\left(-\frac{k^2}{\sigma_c^2}\right) \tag{3}$$

where the parameter σ_c establishes the decrease with the curvature.

In addition, since the connection strengths between cells decay with distance, we also define a distance weight function and then normalize the values,

$$W(x', y'; x, y) = \exp\left[-\frac{(x'-x)^2 + (y'-y)^2}{\sigma_d^2}\right] \quad (x', y') \in A_e \tag{4}$$

$$S = \sum_{(x',y') \in A_e} W(x', y'; x, y) \tag{5}$$

$$W_d(x', y'; x, y) = \frac{W(x', y'; x, y)}{S} \tag{6}$$

where A_e denotes excitatory regions. The parameter σ_d establishes the decrease with the distance.

The excitatory connection strengths depend on curvature and distance. Accordingly, the facilitatory input $F(x, y, \alpha)$ for the cell with the RF center (x, y) and preferred orientation α , coming from its surround, can be expressed as follows,

$$F(x, y, \alpha) = \sum_{(x', y') \in A_e} \sum_{\beta} W_c(x', y', \beta; x, y, \alpha) W_d(x', y'; x, y) R(x', y', \beta) \tag{7}$$

where A_e is the same description as above, $R(x', y', \beta)$ denotes the response of the cell of position (x', y') and orientation β .

2.2 Iso-Orientation Inhibition

The responses of cortical cells to stimuli in the RF are suppressed by their environments, and the degree of suppression depends on the orientation contrast of the surround and the RF stimuli [1]. The inhibitory effect also declines with increasing distance to the RF. We construct a center-surround difference-of-Gaussians (DoG) filter to implement the distance-weighted representation,

$$DoG(x, y) = \frac{1}{\sqrt{2\pi} 4\sigma} \exp\left(-\frac{x^2 + y^2}{32\sigma^2} \right) - \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2} \right) \tag{8}$$

In addition, inhibitory regions whose direction is complementary with excitatory ones (i.e. $\pi - \varphi < |\gamma - \alpha| < \pi$) are located in both sides of the RF with bow-tie-shaped neural connections. The oriented inhibitory regions A_s are relevant to the preferred orientation α of the stimulus within the RF. We normalize the weighting function in the method similar to in [11], and it is defined as follows,

$$W_r(x, y; \alpha) = \frac{N(DoG(x, y))}{\mathbf{Z}} \quad (x, y) \in A_s \tag{9}$$

$$N(DoG(x, y)) = \begin{cases} DoG(x, y) & DoG(x, y) > 0 \\ 0 & else \end{cases} \tag{10}$$

$$\mathbf{Z} = \sum_{(x, y) \in A_s} N(DoG(x, y)) \tag{11}$$

where A_s denotes inhibitory regions, the function $N(\cdot)$ ensures to generate inhibitory action only in the inhibitory regions.

An orientation difference-based weighting function is given by,

$$W_\Delta(\beta; \alpha) = \exp\left[-8 \left(\frac{\theta_\Delta}{\pi} \right)^{1.5} \right] \tag{12}$$

where θ_Δ denotes the orientation difference between A and B, its definition as follows:

$$\theta_\Delta = \min(|\beta - \alpha|, \pi - |\beta - \alpha|) \tag{13}$$

where α and β denote the orientations inside and outside the RF, respectively.

Thus the stimulus with the RF center (x, y) and preferred orientation α suffer the surround suppression $S(x, y, \alpha)$ coming from all the orientations is computed as follows,

$$S(x, y, \alpha) = \sum_{i=1}^{K} W_\Delta(\beta_i; \alpha) s(x, y, \beta_i) \tag{14}$$

where K is the number of sampling orientations, $s(x, y, \beta_i)$ which represents the inhibitory element coming from orientation β_i is computed in the following way,

$$s(x, y, \beta_i) = W_r(x, y; \alpha) * R(x, y, \beta_i) \tag{15}$$

where $*$ denotes the convolution operator, and $R(x, y, \beta_i)$ is as in (7).

2.3 Model Description

The recurrent connections of excitatory and inhibitory neurons indicate that the cortical networks achieve specific visual tasks in a dynamic and flexible fashion [5]. A dynamic model is provided to describe the local interactions in perceptual grouping,

$$R^t(x, y, \alpha_i) = R^{t-1}(x, y, \alpha_i) + \eta(t)\left[F(x, y, \alpha_i) - S(x, y, \alpha_i)\right] \quad i = 1, 2, \cdots K \tag{16}$$

$$\eta(t) = e^{-0.02(t-1)} \tag{17}$$

with K as in (14). The factor $\eta(t)$ gradually decreases with the increasing iterations to ensure that the dynamic process would converge. According to a subjective visual appreciation, we choose Gabor energy filters with a proper center frequency from a series of frequencies to pre-process an input image, and then the corresponding Gabor energy is used to initialize $R^0(x, y, \alpha_i)$. To terminate the iteration, we can define, in advance, either the maximum number of iterations or a lower bound of the change in successive steps.

Finally, a winner-take-all selection procedure is performed, e.g., the maximum response of each pixel over all the orientations as the model's output.

3 Experimental Results

To verify the performance of our algorithm, we carry out the following experiments, including analysis of excitatory lateral connections, analysis of inhibitory lateral connections and effect of applying the model to real images.

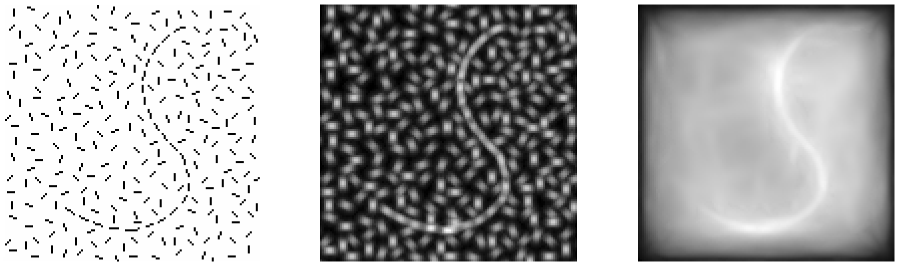


Fig. 2. Enhancement effect. The left, middle, right columns correspond to original image, Gabor energy and output only considering excitatory connections, respectively. (center frequency of Gabor energy is 0.325).

First, we test the ability of the collinear enhancement scheme in (7) to extract coherent spatial configuration from the cluttered background. From fig.2, it can be seen that surround facilitatory actions make cluttered elements tend to be uniform while a well-organized structure pop out perceptually from its background. An additional advantage, some small gaps on the curve are filled in by contextual interactions.

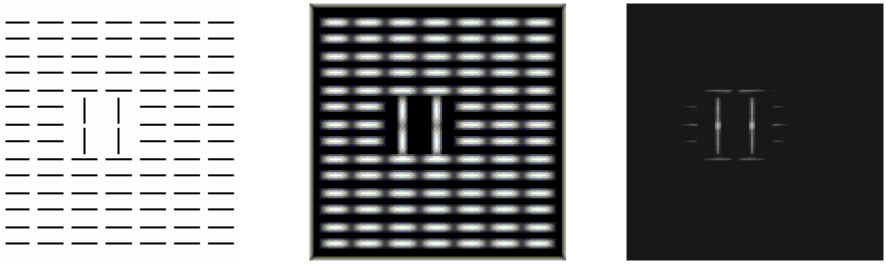


Fig. 3. Suppression effect. The left, middle, right columns correspond to original image, Gabor energy and output only considering inhibitory connections, respectively. (center frequency of Gabor energy is 0.275).

Fig.3 illustrates iso-orientation inhibition effect, which is obtained by only considering inhibitory term. The loss of this suppression in areas where there is a change in orientation results in enhanced saliency of the texture boundary. The proposed scheme shows an excellent performance for this implicit contour extraction.

From the above two examples, we can see that facilitation and suppression play different roles in contour processing— the former is deemed to be important in contour integration and saliency; the latter is thought to be important in the segmentation of surfaces and textures. Both the mechanisms are necessary to contour processing. Our results are in agreement with the perception of visual system.

Finally, we apply the model to real images, as shown in fig.4. The top right image is a model with a stripe coat and camouflage trousers, the middle a snail, the bottom a satellite photo of a river. The contours of these images are embedded in cluttered

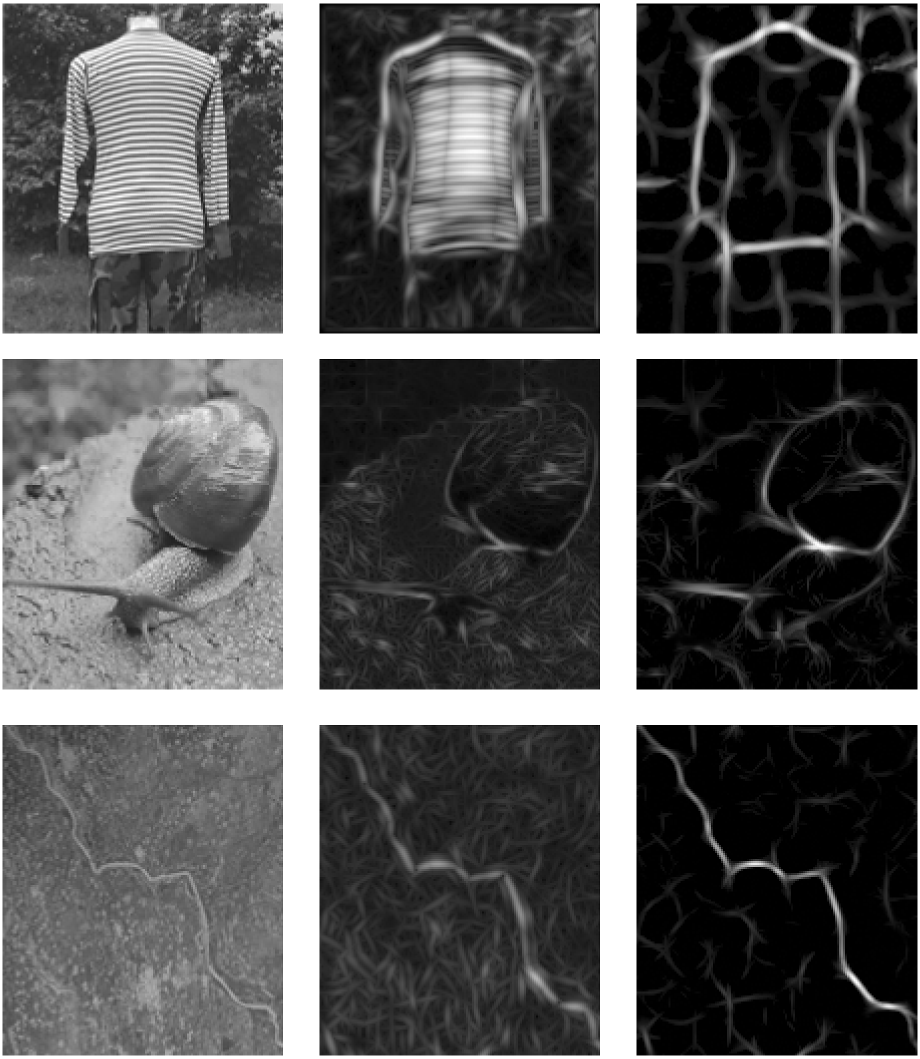


Fig. 4. Results of applying the model to real images. The left, middle, right columns correspond to original image, Gabor energy and outputs of our model. (from top to bottom, center frequency of Gabor energy is 0.225, 0.250, 0.225).

backgrounds, especially, in the first, the object per se has texture. The iso-orientation inhibition dramatically reduces the response of uniform texture and moreover, plays an important role in eliminating the interference of the extraneous elements and preventing them from extending to the contour of object. The collinear excitation strengthens coherent elements and preserves the integrity of the smooth contour.

The results show that the model indeed effectively suppresses texture edges and cluttered elements, and enhances well-organized shape contours.

4 Conclusions

Contour plays a key role in shape-based object detection. Several difficulties lie in drawing the object contour from complex natural scene: I) most local oriented edges engendered by texture must be eliminated while preserving the object border; II) some important shaping contours have not been well defined such as texture boundary. To address this problem, we developed a bottom-up model directly inspired by the long-range neural interactions in primary visual cortex.

Several main contribution of this paper lies in applying perceptual characteristics of prime visual cortex to contour extraction in computer vision, and obtaining satisfactory results in the test for objects with relatively simple and smooth structures. In this process, we put some emphases on the two different roles - facilitation and suppression - played in the contour processing. For the facilitation our facilitatory scheme can better interpret visual perceptual grouping, and contribute to further understand visual mechanisms. For the suppression, we stress the importance of anisotropic inhibition to popping out texture boundaries, which differs intrinsically from the isotropic inhibition adopted in many literatures.

The present work may be further improved by adopting different local scales according to local features of an image, which can more accurately extract local edges and supply more integration information for local element grouping. This is the work we plan to extend.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Contract 60135020.

References

1. Knierim J.J., van Essen D.C.: Neuronal responses to static texture patterns in area V1 of the alert macaque monkeys. J. Neurophysiol. Vol.67 (1992) 961-980.
2. Kapadia M.K., Ito M., Gilbert C.D., Westheimer G.: Improvement in visual sensitivity by changes in local context: parallel studies in human observers and in V1 of alert monkeys. Neuron. Vol.15 (1995) 843-856.
3. Li Z.P.: Visual segmentation by contextual influences via intracortical interactions in primary visual cortex. Comput. Neural Syst. Vol.10 (1999) 187-212.
4. Polat U., Mizobe K., Pettet M.W., Kasamatsu T., Norcia A.M.: Collinear stimuli regulate visual responses depending on cell's contrast threshold. Nature. Vol.391 (1998) 580-584.
5. Bonneh Y., Sagi D.: Effects of spatial configuration on contrast detection. Vis. Res. Vol.38 (1998) 3541-3553.
6. Tang Q.L., Sang N., Zhang T.X.: A neural network model for extraction of salient contours. Wang J. et. al. (eds.): ISNN. LNCS, Vol. 3497. Springer-Verlag, Berlin Heidelberg Chongqing (2005) 316-320.
7. Shouval H.Z., Goldberg D. H., Jones J. P., Beckerman M., Cooper L. N.: Structures long-range connections can provide a scaffold for orientation maps. J. Neuroscience. Vol.20 (2000) 1119-1128.

8. Parent P., Zucker S.W.: Trace inference, curvature consistency and curve-detection. IEEE Trans. Pattern Anal. Machine Intell. Vol.11, No.8 (1989) 823-839.
9. Yen S.-C., Finkel L.H.: Extraction of Perceptually Salient Contours by Striate Cortical Networks. Vision Res., Vol.38, No.5 (1998) 719-741.
10. Hess R.F., Beaudot W.H.A., Mullen K.T.: Dynamics of contour integration. Vision Res. Vol.41 (2001) 1023-1037.
11. Grigorescu, C., Petkov, N., Westenberg, M.A.: Contour Detection Based on Nonclassical Receptive Field Inhibition. IEEE Trans. Image Processing. Vol.12, No.7 (2003) 729-739.

Identification of Printing Process Using HSV Colour Space

Haritha Dasari and Chakravarthy Bhagvati

Dept. of Computer and Information Sciences,
University of Hyderabad, Hyderabad 500046, India
harithadasari@rediffmail.com, chakcs@uohyd.ernet.in

Abstract. Inkjet and laser printers, and photocopiers are being increasingly used in criminal activities such as counterfeiting currency, creating forged documents, illegitimate business transactions and terrorism related acts. Identification of the printing process greatly aids in detecting such activities and is extensively used in the field of document examination. In this paper, we propose the use of image processing techniques in identifying the printing process used to generate a document. The characteristics of the various types of non-impact printing methods used by photocopiers, inkjet and laser printers are studied using colour image processing. HSV color space and, in particular, hue images at high-resolution, distribution of isolated spots in the vicinity of hue edge pixels and periodicities in edge intensity profiles distinguish between the different printing processes. Our initial study indicates their promise in replicating the results traditionally obtained by document examiners using a microscope or through chemical analysis.

1 Introduction

Forensic examination of documents is fast emerging as a challenging field of research with the proliferation of fake and *questioned* documents through the use of computers or computer-based technologies. A document is labeled a *questioned document* if its authenticity is in doubt. A questioned document may be genuine; partially faked by obliterating, erasing or altering the original information; or, completely faked as, for example, in counterfeit currency and lottery tickets, and blank educational certificates.

A useful first step in examining documents is to detect how they are created. Are they printed using an inkjet or a laser printer, or are they photocopies? The examination of printing method is useful in detecting whether a number of documents originated from the same source. It is also useful in tracing the document's source much the same way a bullet may be traced to the gun from which it is fired. It is often possible to identify the make of the printer or the copier and sometimes even the individual machine from a careful examination of the marks left behind on the paper or in the inking of the letters.

The conventional methods used by expert questioned document examiners are varied, sophisticated and sometimes very specific involving both destructive and

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 692–701, 2006.

non-destructive tests. Brunelle[3], Ellen[5] and Hilton[7] offer excellent overviews of such techniques and methods. Some commonly examined physical features are machine defects such as trash marks, pitting in the rollers, broken mechanisms and indentations. Additional information is obtained by chemical analysis of inks and papers.

In 1999 Doherty [4] gave an overview on the state-of-the-art in classification of inkjet printers and inks. Examination under a microscope is a favoured method in distinguishing between photocopiers, laser and inkjet printers. When magnified, inkjet appears as a series of irregular coloured dots at planar level with some peripheral bleeding caused by absorption on the paper. Laser printers and photocopiers scatter toner particles over non-image areas. Unlike in inkjet printing, the image areas do not appear as discrete dots because the toner is heated and pressed into the paper. However, the image appears to be raised due to the deposition of toner on the paper[6].

Printing inks may be distinguished chemically to some extent using Mass spectroscopy and chemical analysis. The binding agents are subjected to Pyrolysis Mass spectroscopy and infrared absorption spectroscopy to isolate the inorganic components which are then identified by emission spectroscopy and microprobe electron microscopy[5]. In 1993 Lofgen[8] worked on HPLC analysis of printing inks as a non-destructive technique. Yair's work[11] on classifying printers using colour registration, resolution, text quality, line quality and dot quality is also often quoted in forensic journals.

The use of image processing techniques in forensic document examination is relatively new[5] and is an exciting application area. The work of Agarwal and others[1] described algorithms to decipher obliterated text and analyse stroke sequences. Our earlier experiments [2] show that the hue, saturation and value histograms are similar for the sample words written by the same writing instrument (pen or printer). We also showed that saturation histograms reveal the difference in absorption characteristics of inks and are thus useful in discriminating between liquid, viscous and powdered inks. The Center for Excellence in Document Analysis and Recognition (CEDAR) at Buffalo is also reporting interesting results on writer identification[10].

In this paper, we study the problem of identifying different printing processes using HSV colour space with emphasis on inkjet and laser prints, and photocopies as they are the most common forms of documents. The basic idea is to replicate the analyses and the features used by document examination experts. We selected HSV colour space as subtle changes in colour are significant in the analysis. HSV space offers several advantages in representing colour for image processing. Colour, given by H and S, is decoupled from intensity and operations may be defined that manipulate colour independent of intensity and vice-versa. Also, MacAdam ellipses[9], representing regions in colour space that contain undistinguishable colours, are more homogeneous and compact in HSV space. It implies that colour distances are simpler to interpret in HSV space. HSV space is also useful in document examination as we found that variations in ink colours are well captured in hue, while absorption of the inks into the paper are reflected in saturation values.

The rest of the paper is organized as follows. Section 2 describes characteristics of different printing processes. Section 3 presents our measures and methods using HSV colour space to measure the characteristics described in Section 2. Results are presented and discussed in Section 4 with conclusions in Section 5.

2 Printing Processes and Their Characteristics

There are two primary characteristics in the analysis of printing methods: the nature of the ink used, and the process by which ink is transferred to the paper. Ink may be solid (as generally used in laser printers and photocopiers), viscous paste (as in ballpens) or liquid (as in laser and inkjet printers). The transfer process may broadly be impact type (as in typewriters) or non-impact type (as in laser and inkjet printers). Most modern solid inks are in the form of fine powders which are fused into the paper by applying heat. Mixture of dyes provide the coloring matter, and an important constituent is the resinous material which serves to bind the ink to the paper. Liquid inks by contrast are usually water soluble and do not have a paste-like consistency but are otherwise similar to viscous inks. Offset printing differs in the use of a plate used for selectively applying liquid ink. It consists of *letter areas* that absorb water and others that do not and, consequently, the lettered areas transfer ink to the paper pressed against the plate.

Inkjet Printers use the “Non Impact” method with liquid inks. They use the principle of spraying tiny spots of ink on to the paper. The spots are aimed at the desired locations by a thermal or a piezo electric system.

Laser printers and photocopiers both use indirect electrostatic imaging and are extremely similar in operation. Lasers use dot matrix grid pattern while forming the images on the drum. A laser beam scans across the surface of the drum, selectively imparting points of negative charge in a grid pattern onto the drum’s surface that will ultimately represent the output image. The selective charging is done by turning the laser on and off as it scans the rotating drum, using a complex arrangement of spinning mirrors and lenses. The faster the laser beam is switched on and off, the higher the resolution across the page. On the other hand, fast switching is not possible in photocopiers as they use ordinary white light and differ from lasers in the absence of the dot matrix pattern. In both types of instruments, the image is transferred on to the paper by applying negative charge that attracts toner to the drum. The toner is transferred to the paper as it rolls under the drum, and heat and pressure are applied to the paper. This melts the toner which contains small amounts of wax and fixes it to the paper. Colour is printed as seperate images or layers by mixing Magenta, Cyan, Yellow and finally black toners.

Several attributes define the text quality of a printed page. The main physical characteristics are character hue, edge smoothness, presence of artifacts, uniformity of area fills, raised letters and indentations. Hue refers to the tone of the colour used to print the character. It does not remain constant but is affected by a number of factors such as the rate of ink flow, ink and paper types, the printing process, and many others. Edge roughness or smoothness is determined by printer resolution, dot placement accuracy and the interactions between the colourant and the paper. Unwanted artifacts such as inkjet spray and laser background scatter leave undesirable toner particles near the printed zones. Non uniformity of area fills can occur in a variety of ways, such as the mottle (light and dark areas) caused by the uneven penetration of ink and deviations from true flat surfaces on the drums and papers. Uneven image formation, and temperature and pressure distributions during transfer stage often appear as uneven gloss, banding and density gradients. All non-impact printing methods deposit ink on the paper resulting in raised letters. An impact method on the other hand leaves behind indentations in the paper corresponding to the impacting objects.

3 HSV Colour Features

A colour image is scanned at a high optical resolution ($\geq$ 1200 dpi) using a regular flatbed scanner of a popular make. The image is then converted into HSV and used in the analysis. The saturation and intensity values are scaled to $0 - 255$ range. Any pixel whose saturation is less than 30 and intensity is greater than 235 is taken as a background pixel, i.e., paper. As we used samples printed on white paper in our initial tests, such a global thresholding scheme proved adequate. The focus of this paper is on methods for identifying printing processes rather than on preprocessing and the separation of foreground text from background paper is not discussed any further.

The raised letters in photocopiers, inkjet and laser printers are revealed in the hue components. As the scanner scans the document in a single direction, the light first strikes the rising edge of the letters and then the falling edges. The movement of the light source in a single direction causes shadows to be formed beyond the falling edges. The increased brightness on the rising edge and the shadows on the falling edges result in a characteristic row-wise hue profile that produces low contrast on the rising edges and high contrast on the falling edges.

The second feature used in discrimination is image overspray/smear. It is a measure of the extraneous ink spots that are adjacent to the actual printed matter. To the unaided eye, it appears as either noise or blurring of the edges of the letters. We measure the amount of extraneous dots in a user defined area located within a specified distance from text lines.

Inkspray and oversmear are identified by performing edge-detection on the hue image. Hue component is convolved with a Sobel edge detector and the result is blurred with a 3×3 averaging window to obtain an edge-detected image. We count the number of edge pixels within a distance d in directions orthogonal to the edge orientations. The higher the number, the greater the inkspray.

A third feature is the presence of serrations caused by charging the drum in a grid pattern in laser printers. We found that the serrations are best revealed in the saturation component of the image. We perform edge detection using the same operation described in the previous paragraph and obtain an edge-detected saturation image. The column-wise edge intensity profiles are periodic reflecting the grid pattern of image formation in laser printers. We tested our methods only on documents containing English text and as the English alphabet contains a large number of vertical strokes, the column-wise profiles were found useful. For other types of printed matter, it may also be necessary to examine the row profiles.

4 Results and Discussion

Document samples were taken from six popular inkjet printers manufactured by Canon, Epson, HP and Lexmark, four colour laser printers by HP, Canon and Xerox, and two colour photocopiers. The scanned documents contained text in either blue or green colour on a white background. These documents are scanned at 1200 dpi (optical resolution) and then converted into HSV format.

Figures 2–3 show the hue components of inkjet original and its photocopy, and a laserjet original and its photocopy (Figure 1). Figures 4–5 show the corresponding hue histograms. First, it may be seen that the hue hiostograms of photocopies are wider and sometimes bimodal. The original inkjet and laser prints result in unimodal and narrower hue histograms. The peaks are also shifted in the photocopies. The primary colours used as toners differ for inkjets, lasers and photocopier, the hues present in the original are approximated by a combination of hues corresponding to the toner in the photocopier. It might be inferred that the toner in the inkjet has a blue primary with a hue value of approximately 220, the laser, a value of roughly 210 and the photocopier, a value of 230. The bimodal nature and the broader histograms of the photocopies may be due to the increased variance in the copy and the interaction between the differing hues of the toners that produced the original and its copy.

The appearance of raised letters is clearly seen for laser printers and photocopiers in Figures 2 and 3. It may also be seen that the contrast is lower on the left side of the strokes comprising the letters for the photocopied sample in Figure 2 (note especially the letter ‘i’). For the laser print in Figure 3, it is the left side that has greater contrast (again best seen for the letter ‘i’). One may conclude that the direction of motion of illuminating source is in opposing directions for the particular laser printer and the photocopier used in our testing. It may offer a novel method of identifying the printer model because different printers use different paper paths during the printing process.

Figures 6 – 7 show edge-detected hue components of the images in 1. It may be clearly seen that ink spray is significant in inkjet printing. Laser prints show the least amount of inkspray and it is also restricted to tight boundaries from the actual lettering. Photocopiers show more overspray than laser prints and it is also more distributed. However, an interesting feature is that the inkspray is less in the photocopied inkjet print than in the original (Figure 6). It is possible that the photocopier does not have sufficient sensitivity and resolution to copy all the isolated dots found in the original inkjet print. Therefore, a moderate

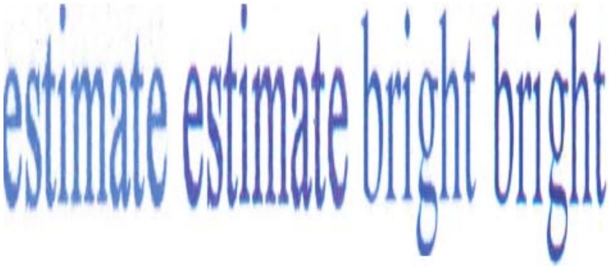


Fig. 1. Samples of inkjet printing, its photocopy, laser printing and its photocopy

amount of inkspray in conjunction with an absence of raised letters may indicate a photocopy of an original inkjet printed document.

Figures 8 – 9 show edge-detected saturation images. The serrated edges of a laser printer are clearly visible in the saturation images. The inkjet and photocopied images do not show such a striking pattern. We plot the column-wise intensity profiles for the edge-detected images. The serrated edges are shown by periodicities in the intensity profiles.

A quantitative measure of periodicity may be derived from Time-Series analysis. A moving average measure may be subtracted from the intensity profile, and the variance of the resulting residual data is a measure of the periodic structure. A high variance indicates periodicity. The period for computing the moving average is given by the grid resolution of the laser printer.

Our initial testing and analysis on blue-coloured text gave very good results on nearly 100 sample images. We replicated the common features, obtained from microscopic examination and variety of chemical and physical analyses by document examiners, using image processing techniques. A new promising result, although not yet conclusive given the small sample size used in initial testing, is a method to identify the relative direction of movement of paper with respect to the illumination source. Such information has a great significance in narrowing down the identification of the printer used to print the document. The method, by itself, may not be robust as the original document may be placed upside-down or in other orientations. In conjunction with the other tests, however, it may provide strong corroborative evidence. Table 1 summarizes our approach.

The experiments are recently extended to documents containing text in green and other colours on a white background and those obtained from two more photocopiers. The results obtained are in accordance with the analysis and conclusions we made in case of blue-coloured text. Examples (Figures 10 and 11) show the similarities in results. First, it may be seen that the hue hiostograms of photocopies are wider and sometimes bimodal. The hue histograms are narrower in original inkjet and laser prints. The peaks are also shifted in the photocopies. The edge-detected saturation images show serrated edges in laser printer output and their distortion in other outputs. There is, thus, a significant potential for applying better image processing techniques for forensic examination of documents.

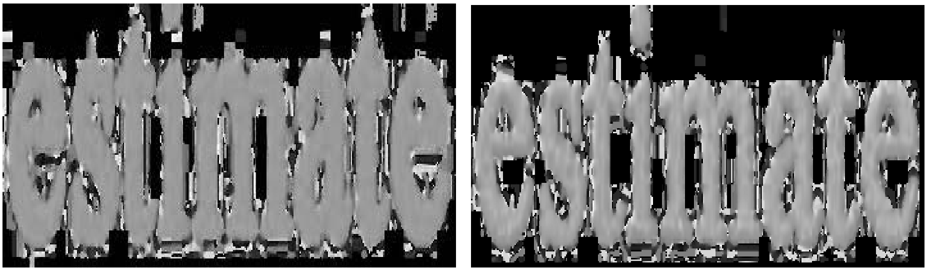


Fig. 2. Hue image of inkjet (left) and photocopied inkjet (right) print samples

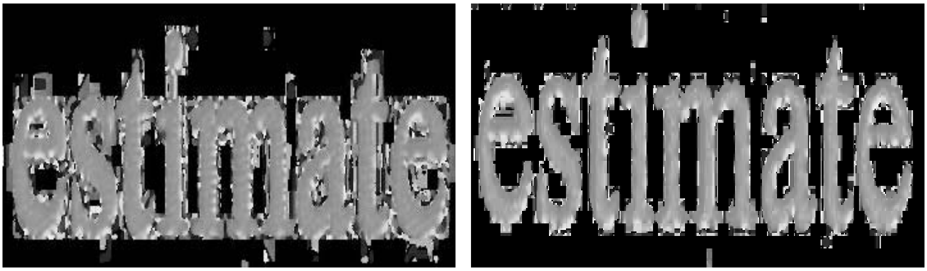


Fig. 3. Hue image of laser (left) and photocopied laser (right) print samples

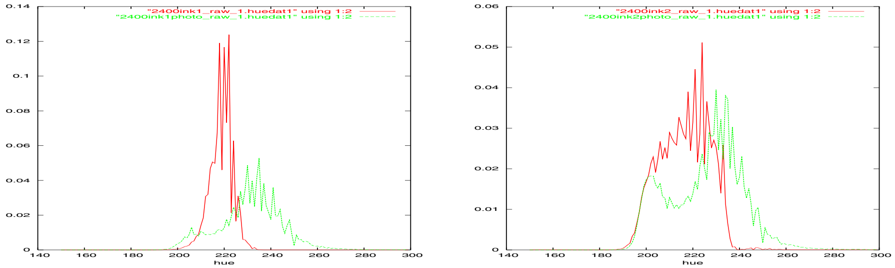


Fig. 4. Hue histograms of inkjet sample and its photocopy for two different printers

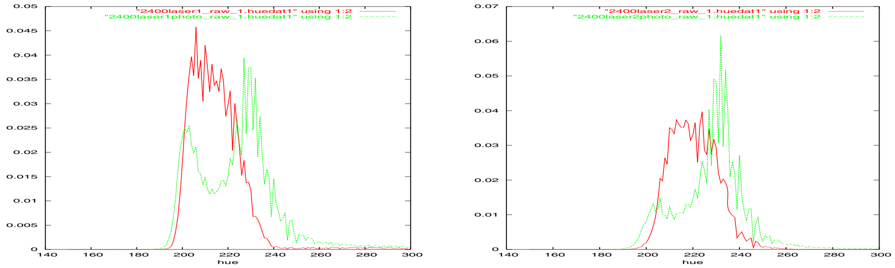


Fig. 5. Hue histogram of laser sample and its photocopy for two different printers

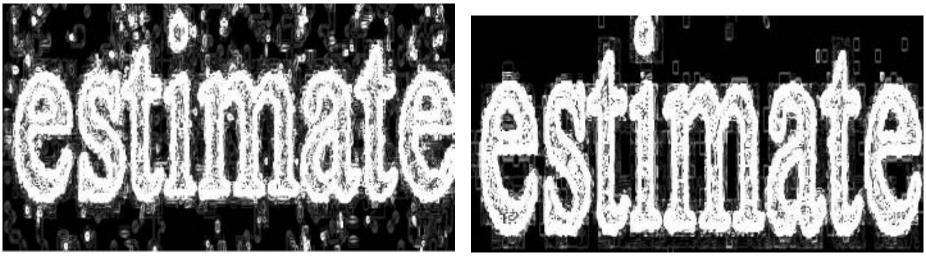


Fig. 6. Edge-detected hue image of inkjet (left) and photocopied inkjet (right) samples

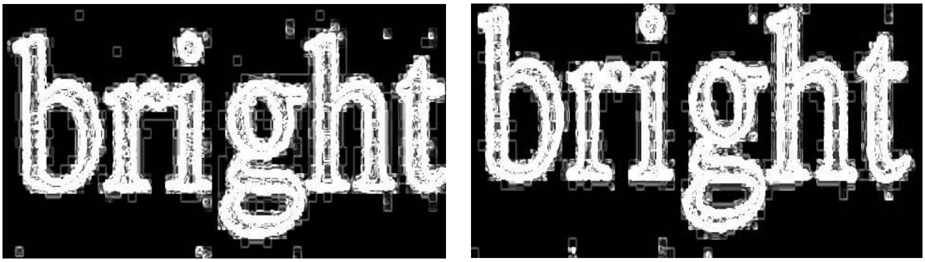


Fig. 7. Edge-detected hue image of laser (left) and photocopied laser (right) samples

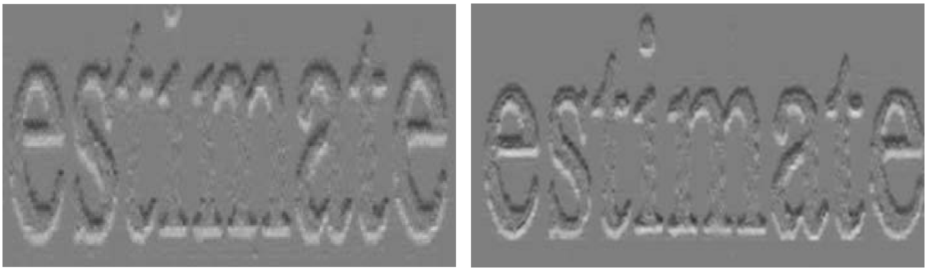


Fig. 8. Edge detected saturation image of inkjet (left) and photocopied inkjet (right) print samples

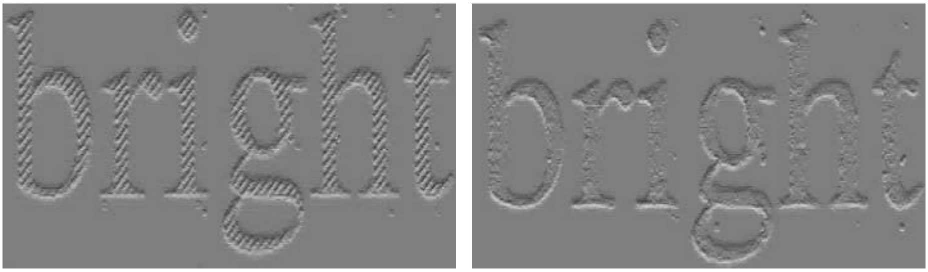


Fig. 9. Edge detected saturation image of laser (left) and photocopied laser (right) print samples

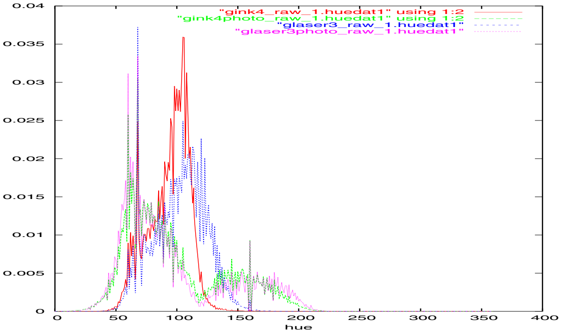


Fig. 10. Hue histograms of inkjet, laser and corresponding photocopied samples for green-coloured text

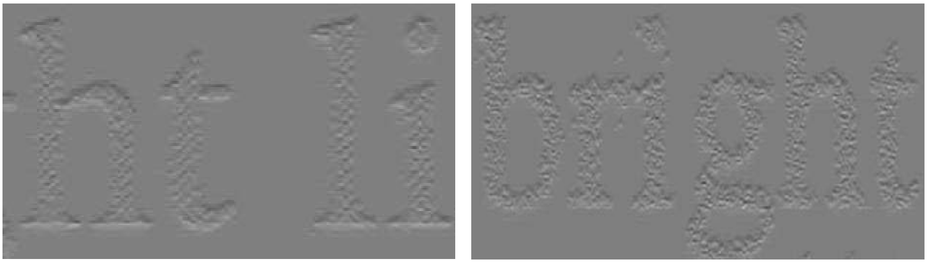


Fig. 11. Sobel edge detected saturation image of laser (left) and photocopied laser (right) print samples for green coloured text

Table 1. HSV features for identifying printing processes

Printing Process	HSV Features
Original Inkjet Print	Narrow unimodal hue histogram, large number of isolated dots near the strokes and no variation in contrast on opposite sides of strokes in edge-detected hue image
Original Laser Print	Narrow unimodal hue histogram, small number of isolated dots near the strokes; alternating low and high contrasts on opposite sides of strokes in edge-detected hue images; and, periodic variation in column-wise intensity profiles in edge-detected saturation images
Photocopied Inkjet Print	Wide, bimodal hue histogram; small number of isolated dots near the strokes; and, no periodicity in column-wise intensity profiles in edge-detected saturation images
Photocopied Laser Print	Wide, bimodal hue histogram; large number of isolated dots near the strokes; and, irregular variations in column-wise intensity profiles in edge-detected saturation images

5 Summary and Conclusions

In this paper, we have shown that features derived from HSV colour space are useful in forensic examination of documents. In particular, we showed that hue and saturation components reveal vital information about the printing processes used in generating a document. We replicated the most common features used traditionally by document examiners and also potentially discovered a new method to identify printer model based on the relative direction of movement of paper with respect to the illuminating source for laser printers and photocopiers. In conjunction with the results on identifying ink types, the results in this paper present a powerful set of tools to assist a document examiner in detecting forgeries. Finally, we suggest that developing algorithms and techniques for use by document examiners is an exciting area of research for document image processing community in addition to the traditional applications of preprocessing images and extracting features for use in optical character recognition.

References

1. Agarwal A., Chakravarthy B., Jain R.K., Rao M.S. Computer based Decipherment of Obliterations in Questioned Documents, in *Proc. National Conference on Document Analysis and Recognition (NCDAR2003)*, Mandya (2003).
2. Haritha, D., Chakravarthy Bhagvati. Classification of Liquid and Viscous inks using HSV Colour Space, *Int. Conf. on Document Analysis and Recognition (ICDAR2005)*, Seoul, Korea (2005).
3. Brunelle, R. Questioned Document Examination in Forensic Science Handbook, in *Applied Optical Journal*, Vol 5, (1966), pp 1361–1364.
4. Doherty, P.E. The Classification of Inkjet Printers and Ink. in *Proc. 57th ASQDE Annual meeting*, (1999).
5. David Ellen. *The Scientific Examination of Documents*, Taylor & Francis, 2nd Edition,(1997).
6. Gerald M. Laporte, Robert S. Ramotowski. The Effects of Latent Print Processing on Questioned Documents Produced by Office Machine Systems using Inkjet Technology and Toner, *Journal of Forensic Science*, Vol 48, (2003).
7. Ordway Hilton. *Scientific Examination of Questioned Documents*, CRC Press, (1993).
8. Lofgen B, Andrasko J. HPLC Analysis of Printing Inks. *Journal of Forensic Sciences*, Vol 38, (1993), pp 1151-1154.
9. MacAdam, D.L. *Sources of Color Science*, MIT Press, Cambridge, MA, USA, (1970).
10. Srihari S.N., Huang C., Srinivasan H., Shah V.A. Biometric and Forensic Aspects of Digital Document Processing *Digital Document Processing*, B.B.Chowdary(ed.), Springer (to appear).
11. Yair kipman. Image Quality Metrics for Printers and Media. *Proc. IS & T's PICS conference*, (1998).

Spatiotemporal Density Feature Analysis to Detect Liver Cancer from Abdominal CT Angiography

Yoshito Mekada[1], Yuki Wakida[2], Yuichiro Hayashi[2],
Ichiro Ide[2], and Hiroshi Murase[2]

[1] School of Life System Science and Technology, Chukyo University,
101 Tokodachi Kaizu Toyota, 470-0393, Japan
y-mekada@life.chukyo-u.ac.jp
http://www.st.chukyo-u.ac.jp/y-mekada/
[2] Graduate School of Information Science, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan
yhayashi@suenaga.m.is.nagoya-u.ac.jp, ide@is.nagoya-u.ac.jp,
murase@is.nagoya-u.ac.jp

Abstract. In this paper, we propose a method of detecting liver cancers from dynamic X-ray computed tomography (CT) images based on a two-dimensional histogram analysis. In the diagnosis of a liver, a doctor examines dynamic CT images. These consist of four images, namely the pre-contrast phase, early phase, portal phase, and late phase ones, which are taken sequentially within a few minutes. Since the early and late phase images are important for diagnosing liver cancer, our method refers to both of them for detecting suspicious regions and eliminating false positives. First, it extracts liver cancer candidates by applying an adaptive neighbor type filter to the late phase image. Then, precise cancerous regions are specified by a region forming method. Most of the false positive regions are eliminated by two-dimensional histogram analysis of each region of interest. We applied the proposed method to 21 dynamic CT images. The results showed that sensitivity was 100% and there were 0.33 false positives per case on average.

1 Introduction

As a result of recent progress in computed tomography (CT) imaging devices such as multi-detector row CT(MDCT) scanners, these devices now generate a huge number of high-resolution slice images of a patient. In particular, in diagnosis using CT angiography, a doctor examines two or more three-dimensional images. For example in liver diagnosis, four CT images taken at different phases after the injection of a contrast medium are used routinely. Studying these images imposes a heavy load on medical doctors. Therefore, a computer aided diagnosis system is required.

In this paper, we propose a method of detecting liver cancer from CT angiography by using a registration technique and a two-dimensional histogram analysis of registered images. There has been some research on computerized

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 702–711, 2006.

detection of cancerous regions from CT angiography [1-4]. Masumoto et al. [1] developed the image features for detecting cancer lesions from a single phase image, and Hong et al. [2] studied the same kind of scheme for three phase images independently. Watanabe et al. [3] proposed two kinds of density transition features to enhance the cancerous region. Their system needs four phase images that must be registered beforehand. Thus, their system suffers from the limitations of the medical situation. That is, in liver diagnosis, only the early and late phase images are frequently taken to reduce radiation exposure. Shimizu et al. [4] used the early and late phase images that were aligned using a non-rigid registration technique based on free-form deformation. They enhanced cancerous regions from each phase image independently by using an adaptive convergence index filter [5] and integrated these enhanced images into one image using addition operations. Some of these systems are sensitive to the image conditions and do not treat an important medical finding of hepatocellular carcinoma (HCC) mentioned in the next section.

The proposed system consists of three parts: detection of cancerous regions from late phase images, estimation of region borders, and false positive reduction based on two-dimensional histogram analysis to quantify important medical findings.

2 Characteristics of Liver Cancer

The early phase image is acquired shortly after the injection of contrast medium, and the late phase image is taken several minutes later. In a CT image, the contrast medium makes the CT value high because it absorbs more X-rays than the abdominal organs. In the early phase image, the contrast medium accumulates in the hepatic artery. It also accumulates in the HCC. On the other hand, in the late phase image, the contrast medium does not accumulate in any specific organ, but is distributed uniformly among all the organs. However, within an HCC lesion, the CT value is lower than that of surrounding tissues because the contrast medium had flowed out by that time.

Examples of each phase image are shown in Fig.1. From these figures, it looks easy to detect cancer regions from early phase images by using a simple thresholding technique. However, it is difficult to separate an HCC lesion from the hepatic artery because most HCCs are connected to the artery. On the other hand, in the late phase image, while differences in CT value between an HCC

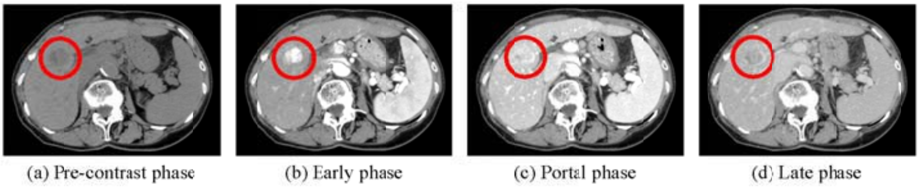

(a) Pre-contrast phase
(b) Early phase
(c) Portal phase
(d) Late phase

Fig. 1. Examples of CT angiogram

lesion and surrounding tissues are relatively small, the lesion border stands out clearly. Using these image features, we detect HCCs in the following way.

1. Partially detect a cancerous region from late phase images by spatial density analysis.
2. Estimate its border from late phase images by modifying the region forming method from partial borders.
3. Eliminate false positive regions by two-dimensional histogram analysis from early and late phase images aligned by a non-rigid registration technique.

3 Method

The processing flow of our method is shown in Fig.2. The inputs are the early and late phase images and the liver region in the late phase image detected by the method given in ref. [6].

3.1 Detection of Cancer Candidates

We evaluated the features of cancer lesions in late phase images, where the CT value of a cancer lesion is lower than that of surrounding tissues. It is difficult to extract cancerous regions using a simple thresholding technique because there is no significant difference in CT value between a cancerous region and non-cancerous liver tissues. Therefore, we extracted hollows from the late phase image by applying an adaptive neighbor type filter. Then, precise cancerous regions were specified by a three-dimensional region forming method.

Finding Cancer Candidates. The adaptive neighbor type filter involves the following steps.

1. Set 26 search directions for an arbitrary voxel $\boldsymbol{x} = (i, j, k)$ in the liver region. The three components of direction vector $\boldsymbol{d}_n(|\boldsymbol{d}_n| > 0, n = 1, 2, \ldots, 26)$ are defined by all combinations of -1, 0, 1.
2. Find the nearest voxel that satisfies at least one of the following conditions for every n-th $(n = 1, 2, \ldots, 26)$ direction.
   (C1) $f(\boldsymbol{x} - r\boldsymbol{d}_n) - f(x) > T_1, r = (1, 2, \ldots, L_{max})$
   (C2) $h(\boldsymbol{x} - r\boldsymbol{d}_n) = 0, r = (1, 2, \ldots, L_{max})$,
   where $f(\boldsymbol{x})$ is the CT value at voxel $\boldsymbol{x}$ of the late phase images, h is a binary image whose value is 1 for the liver region detected by the method in [6] and 0 outside the liver region. L_{max} and T_1 are predefined threshold values. Only if the farthest voxel to $\boldsymbol{x}$ satisfying at least one above mentioned conditions is found in the n-th direction, we call this voxel a reach voxel R_n (see Fig.3).
3. Set the value of voxel $\boldsymbol{x}$ to 1 only if all the following three conditions are satisfied.
   (C3) Reach voxels are detected in all directions.
   (C4) The number of reach voxels located outside the liver region is smaller than a predefined threshold value T_2.

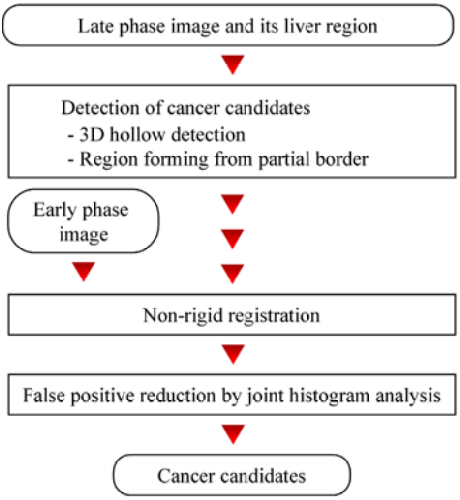

Late phase image and its liver region
Detection of cancer candidates
- 3D hollow detection
- Region forming from partial border
Early phase image
Non-rigid registration
False positive reduction by joint histogram analysis
Cancer candidates

Fig. 2. Flow of the proposed method

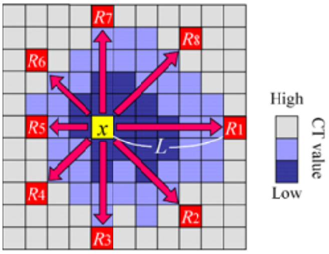


Fig. 3. Illustration of reach point

(C5) At least one reach voxel is located within the liver region from all combinations of two reach voxels that have point symmetry to voxel $\boldsymbol{x}$.

The concept of this filter is similar to the Radial Reach Filter (RRF) proposed by Satoh et al. in [7]. The RRF is a two-dimensional robust gray value change detector for temporal subtracted images, while the filter that we propose is a three-dimensional hollow detector that is sensitive to changes in ambiguous gray values. Conditions (C2), (C4), and (C5) are taken into account to detect cancer lesions at the border of the liver for practical reasons.

Extraction of Precise Region of Cancer Candidates. In this section, we describe a technique for forming the three-dimensional border from partial border voxels. Using the above procedure, we can detect the central part of cancer lesions and some false positive regions. To eliminate these false positive regions by the histogram analysis (explained later), it is important to extract the precise cancerous region. However, it is difficult to detect the actual lesion border because of the unevenness of the contrast medium, unevenness of the cancer lesion itself, and the effect of the contacting tissues. Incidentally, reach voxels tend to be located close to the region border like the RRF [7]. The results of detected cancer candidates and the frequency image of reach voxels are shown in Fig.4. However, voxels that are frequently selected as cancer candidate voxels do not always form a closed surface if there are no significant differences at the partial lesion border. To cope with such situations, we extend the region forming method from the partial borders proposed by Sonka et al. [8] as follows.

1. Count the number of times that each voxel is selected as a reach voxel.
2. For all voxels $\boldsymbol{x}$ among cancer candidates, for all twenty-six directions, mark voxels located between $\boldsymbol{x}$ and the selected voxels as reach voxels most frequently.

3. Compute the number of times each voxel in the image is marked. Let $b(\boldsymbol{x})$ be the number of times voxel $\boldsymbol{x}$ is marked. If $\boldsymbol{x}$ is marked from a single direction, then $b(\boldsymbol{x})$ is set to zero.
4. The weighted number of marked $B(\boldsymbol{x})$ is determined as follows:

$$B(\boldsymbol{x}) = \begin{cases} 0 & for\ b(\boldsymbol{x}) = 0 \\ 1/27 & for\ b(\boldsymbol{x}) = 1 \\ 2/27 & for\ b(\boldsymbol{x}) = 2 \\ 5/27 & for\ b(\boldsymbol{x}) = 3 \\ 10/27 & for\ b(\boldsymbol{x}) > 3 \end{cases}$$

5. Compute $B_m(\boldsymbol{x})$ that is a median of $B(\boldsymbol{x})$ among the $3 \times 3 \times 3$ neighborhood of voxel $\boldsymbol{x}$.
6. Set the value of voxel $\boldsymbol{x}$ to one as a component of a cancer candidate if the sum of $B_m(\boldsymbol{x})$ of the $3 \times 3 \times 3$ neighborhood voxels is greater than 1.0, otherwise mark it as background.

Fig.5 shows the result of region forming method mentioned here.

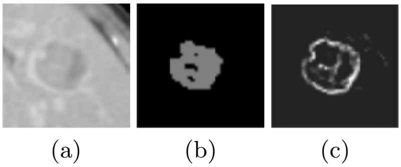


Fig. 4. Extracted cancer candidate and frequency image of reach voxels. (a) Magnified view of cancer lesion in late phase image. (b) extracted cancerous voxels by 3.1.1. (c) frequency image of reach voxels.

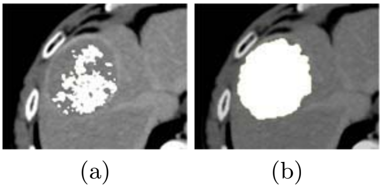


Fig. 5. Example of extracted cancer candidate after 3D region forming method. (a) extracted cancerous voxels by 3.1.1. (b) extracted results by region forming method.

4 Image Registration

The liver is an elastic organ, so it is easily deformed by the patient's respiration, and it also moves along with body movement. Since there is an interval of several minutes between the taking of early and late phase images of CT angiography, non-rigid registration is required for the evaluation of CT value transition. To align the early and late phase images, we used non-rigid registration using the free form deformation technique proposed by Rueckert et al. [9]. This non-rigid registration uses normalized mutual information as a similarity criterion. The early phase image is deformed based on the B-spline interpolation. Fig.6 shows examples of the integrated pre-registration of early and late phase images and the results of non-rigid registration of the two images. The chess-board visualization technique was used to show the results. Brighter rectangles correspond to the late phase image.

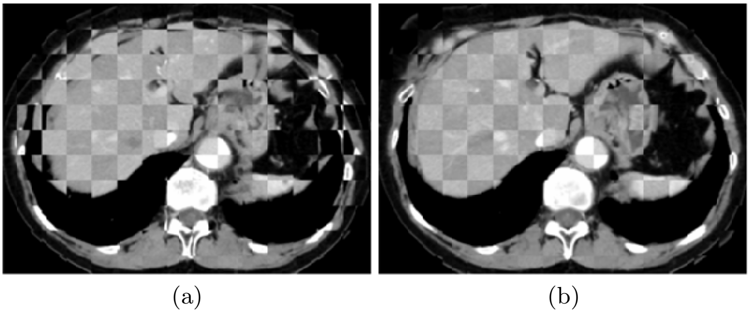


Fig. 6. The chess-board visualization of (a):pre-registration images and (b):the results of non-rigid registration

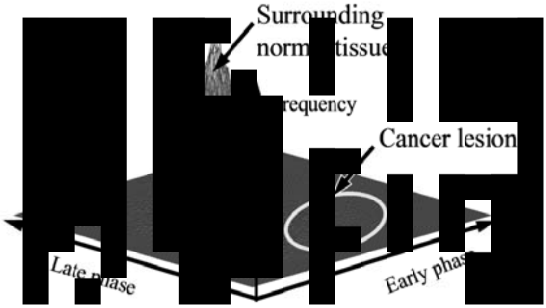
5 Elimination of False Positives

As mentioned in section 2, the CT value of the cancer lesion is higher than that of surrounding tissues in the early phase image and lower in the late phase image. False positive regions are eliminated by evaluating this phenomenon through the following procedure.

1. Definition of region of interest (ROI)
   The ROI is defined for each cancer candidate in order to include normal liver tissue around the cancer candidate region (CCR). The shape of the ROI is a rectangular parallelepiped, whose edges are parallel to the image axes. It is similar to the rectangular parallelepiped circumscribing each CCR. The ROI's center of gravity coincides with that of the CCR, and the volume of the ROI is determined to be 20 times that of the CCR. The volume of voxels outside the liver region is not taken into account in the ROI's volume.
2. Making a joint histogram of the ROI
   A joint CT value histogram of the ROI obtained using registered early and late phase images is calculated. This two-dimensional histogram has one large peak corresponding to the normal tissues around the CCR (see Fig. 7).
3. Estimation of normal tissue distribution
   In this step, the distribution of CT values of normal tissues around the CCR is estimated from the joint histogram. Most of the voxels in the ROI correspond to normal tissues. The joint histogram is projected onto each axis by the maximum intensity projection method. Then, the distribution of normal tissues is defined by the intersection of the following two closed intervals.

   $$[\mu_x - 0.5\sigma_x, \mu_x + 0.5\sigma_x],$$
   $$[\mu_y - 0.5\sigma_y, \mu_y + 0.5\sigma_y],$$

   where μ_x and σ_x are the average and standard deviation, respectively, of the curve projected onto the early phase axis, and μ_y and σ_y are those for the late phase axis.



[redacted]

Fig. 7. Example of joint histogram

4. Estimation of the lesion's distribution
   Because the CT value of a cancer lesion is higher than that of normal tissues in the early phase image and lower in the late phase image, voxels satisfying both of the following conditions are detected as the cancerous region.
   - CT value of early phase is higher than $\mu_x + 0.5\sigma_x$
   - CT value of late phase is lower than $\mu_y - 0.5\sigma_y$

   Let H be the number of voxels in each CCR.
5. Elimination of false positives
   If H is small relative to the CCR volume, such a CCR might be a false positive region. Actually, all CCRs that satisfy the following condition are eliminated as false positives.
   $H/(S - C) \times 100 < T_3$,
   where S is the volume of CCR and C is the volume of cyst in the CCR. The CT value of the cyst is relatively low and hardly changes in the CT angiogram. In this experiment, voxels with a CT value lower than 50 [H.U.] were attributed to the cyst in the CCR.

6 Experiment

The procedure mentioned above was applied to 21 cases of multi-phase abdominal X-ray CT images. Every image was a slice with a size of 512 × 512 [points] having spatial resolution of about 0.6 [mm]. Nineteen cases were taken from a 4-line MDCT device and had a beam thickness of 2 [mm] and reconstruction interval of 1 [mm]. The other two cases were taken from a 16-line MDCT device

Table 1. Image specification of CT images

Image size	512 × 512
# of slices	161∼464
Pixel size[mm]	0.546∼0.625
Reconstruction pitch[mm]	0.5∼1.0
Slice thickness[mm]	1.0∼2.0

and had a beam thickness of 1 [mm] and reconstruction interval of 0.5 [mm]. Seventeen cases included one or more liver cell cancers and the other four cases had no cancer lesions. The image specifications are shown in Table 1. Threshold values were set to $T_1 = 15, T_2 = 13$,and $T_3 = 16$, experimentally.

7 Results

The experimental results show that the average number of false positives per case was 0.33 with sensitivity of 100%. This result is promising because it shows fewer false positives per case for the same database than refs. [3] and [4], which had 0.71 and 0.53, respectively. Fig. 8 shows a free-response receiver operating characteristic(FROC) curve drawn by changing the threshold value T3. Some detection results are shown in Fig. 9 and 10. False positives that still remained were located at the border of the liver. The cancer lesion corresponding to the minimum value of the false positive elimination criterion makes it difficult to improve accuracy. In this experiment, extraction of the precise border did not work well because this lesion was surrounded by many cysts.

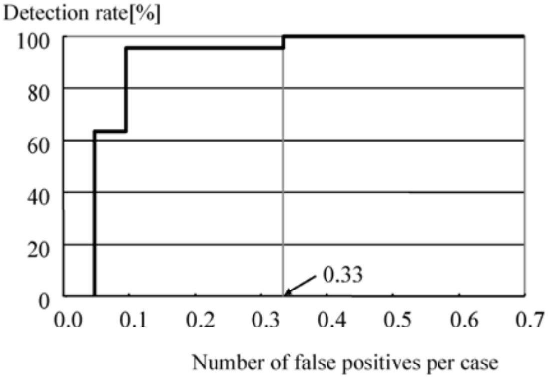


Fig. 8. The FROC curve drawn by changing the threshold value T_3

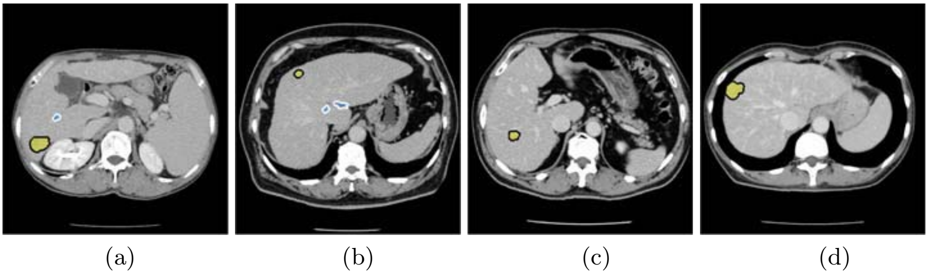
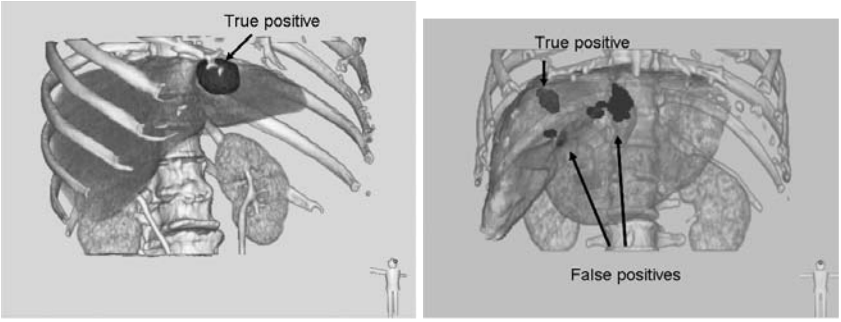


Fig. 9. Examples of detection results. Regions surrounded by the black line are the cancer lesions detected correctly. Regions surrounded by the white line are false positives eliminated correctly.



(a) (b)

Fig. 10. Volume rendering views of results. Yellow regions are the cancer lesions detected correctly. Blue regions are false positives eliminated correctly.

8 Conclusion

This paper presented a novel computer aided liver cancer detection system for three-dimensional CT angiogram. The concept consists of the following.

- Integration of the early and late phase images by non-rigid registration
- Quantification of the density transition state, which is an important diagnostic finding, for each cancerous region detected by spatial density analysis

First, spatial density analysis is used to detect ambiguous three-dimensional hollows as cancer candidates and form a precise region from partial border voxels. Then, spatiotemporal density feature analysis of the two aligned images (early and late phase images) is performed to eliminate false positives. Experimental results showed that the number of false positive regions per case was 0.33 with sensitivity of 100%. This is superior to previous work using the same image database [3, 4]. In the future, we plan to perform fine registration for each ROI to eliminate false positives and develop some new features to reduce false positives.

Acknowledgements. Authors thank to the members of Murase Laboratory of Nagoya University for useful discussions. Parts of this research were supported by the 21st century COE program, the Grant-In-Aid for Scientific Research from Japan Society for Promotion of Science, the Grant-In-Aid for Cancer Research from the Ministry of Health, Labor and Welfare of Japan, the Grant-in-Aide for Private University High-Tech Research Centre from Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. J. Masumoto M. Hori, Y. Sato, T. Murakami, T. Johkoh, H. Nakamura and S. Tamura, “Automated detection of liver tumors in X-ray CT images”, IEICE Trans. J83-D, No.1, pp.219–227, 2000.(in Japanese)
2. J. Hong, T. Kaneko, R. Sekiguchi, Kh. Park, “Automatic liver tumor detection from CT”, IEICE Trans. Inf. Syst. E84-D No.6, pp.741–748, 2001.

3. S. Watanabe, Y. Mekada, J. Hasegawa and J. Toriwaki,“Liver cancer detection by using transition features obtained from multi-phase CT image”, Proceedings of SPIE, Vol.5747, pp.783–789, 2005.
4. A. Simizu, T. Kawamura and H. Kobatake, “Proposal of computer-aided detection system for three dimensional CT images of liver cancer”, Proc. of Computer Assisted Radiology and Surgery (CARS2005) 19th International Congress and Exhibition, International Congress Series 1281, pp.1157–1162, 2005.
5. H. Kobatake, W. Jun, Y. Yoshinaga, Y. Hagihara and A. Shimizu , “Nonlinear adaptive convergence index filter and their characteristics”, Proc. of ICPR2000, Vol.3, pp.526–529, 2000.
6. Y. Hayashi, D. Deguchi, K. Mori, Y. Mekada, Y. Suenaga and J. Toriwaki, “Development of a method for automated liver region extraction from contrasted 3D abdominal X-ray CT images”, Journal of Computer Aided Diagnosis of Medical Images, Vol.8, No. 1_3, pp.18–30, 2004.(in Japanese)
7. Y. Satoh, H. Tanahashi, C. Wang, S. Kaneko, Y. Niwa, K. Yamamoto, “Robust Event Detection by Radial Reach Filter”, Proc. of 16th ICPR, Vol.2, pp623–626, 2002.
8. M. Sonka, V. Hlavac, R. Boyle, “Image Processing, Analysis, and Machine Vision”, PWS Publishing, pp.174–176, 1998.
9. D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, “Nonrigid Registration Using Free-Form Deformations: Application to Breast MR Images”, IEEE Trans. Medical Imaging, Vol.18, No.8, pp.712–721, 1999.

Fast Block Matching Algorithm in Walsh Hadamard Domain

Ngai Li, Chun-Man Mak, and Wai-Kuen Cham

Department of Electronic Engineering, The Chinese University of Hong Kong

Abstract. A fast block matching algorithm, namely Fast Walsh Search, is proposed for motion estimation in block-based video coding. In our approach, target blocks in current frame and their candidates in reference frame are projected onto Walsh Hadamard domain, allowing early rejection of mismatch candidates to reduce computation requirement. Moreover, we introduce a new method called block pyramid matching that re-uses many previous calculations to further lessen the computation load of our approach. Experimental results show that the proposed algorithm can achieve more accurate motion estimation than the popular three-step-search and diamond search with slight increase in computation requirement only.

1 Introduction

Most video coding standards use motion compensation to reduce temporal redundancy. Motion compensation requires block matching which is to find a matching block in the reference frame that is close to the target block in the current frame. The displacement vector of the matching block is called motion vector, therefore block matching is also called motion estimation. Full search block matching (FSBM) algorithm exhaustively searches through all possible locations in the search window to obtain the matching block that has the least matching error with the target block. However, the computation requirement of FSBM is too high for real-time applications. Fast search algorithms such as three-step search (TSS) [4], four-step search (FSS) [5], new three-step search (NTSS) [6], and diamond search (DS) [7] were developed, which can reduce the computation time significantly at the cost of higher matching error. These algorithms find the minimum error using a gradient-descent approach which implicitly assumes that there is no local minimum.

Recently developed video coding standards such as H.264/AVC [8] use Walsh Hadamard Transform (WHT) to compress DC coefficients. Meanwhile, Hel-Or et al. [1, 2] proposed a real time pattern matching algorithm which works in the Walsh Hadamard (WH) domain. Their matching algorithm first computes a distance using a few WHT coefficients to perform early rejection of mismatch patterns and then focus on a small number of remaining candidates that are more likely to be a correct match of the pattern. Their proposed algorithm reduces computation overheads in WHT by an efficient pruning algorithm in which the intermediate data is effectively exploited.

Motivated by [1, 2, 8], we propose a "Fast Walsh Search" (FWS) that performs block matching in the WH domain. Although it is straightforward to perform motion estimation in spatial domain [4, 5, 6, 7], our proposed algorithm requires only slightly

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 712–721, 2006.

more computation than that of TSS and DS and achieves a more accurate block matching in terms of mean square error (MSE). The high efficiency is because of the pattern matching algorithm suggested by Hel-Or et al. [1, 2] as well as a new matching technique called block pyramid matching (BPM).

This paper is organized as follows. Section 2 introduces the proposed fast motion estimation algorithm in WH domain. Section 3 describes the proposed block pyramid matching. Experimental results and conclusions are given in the Sections 4 and 5 respectively.

2 Fast Block Matching in Walsh Hadamard Domain

2.1 Walsh-Hadamard Transform

WHT BPs contain only ±1 and so the projections of a block of pixels on 2D WH domain require additions and subtractions solely. A particular 2D WHT coefficient of a $k{\times}k$ block is obtained by projecting the block on the corresponding $k{\times}k$ WHT BPs where k=2^n, $n\in\mathbf{Z}^+$. In the following, we shall represent a $k{\times}k$ BP by a vector $\mathbf{h}_{(\mathrm{m,n})}$ in $\Re^{k\times k}$ where m and n are the number of zero-crossing in horizontal and vertical direction respectively. The BPs of an 8×8 block are shown in Fig. 1.

In our approach, we follow the same zigzag path as in [1, 2] where the projections on WHT BPs are performed in increasing sequency (the number of zero-crossings along rows and columns) order. In general, the energy of WHT coefficients of an image decreases along the zigzag order [2, 3]; therefore the projections onto the first few WHT BPs capture a large proportion of information of an image. Hel-Or et al. has utilized this energy packing property in his fast pattern matching algorithm [1,2] which is a pruning algorithm that re-uses many intermediate results to further reduce the computation requirements of the WHT. In this paper, we adopt the same idea to develop a block matching algorithm in WH domain.

2.2 Proposed Block Matching System

Motion vector estimation is an important step in video compression. Motion vectors can be estimated by block matching algorithms that minimize a measure of matching error. Suppose the matching error between the target block at position (x,y) in the current frame F_c, and the reference block at position (x+u, y+v) in the reference frame F_R is E(u,v). The motion vector $(\hat{u},\hat{v})$ is defined as:

$$(\hat{u},\hat{v}) = \arg \min_{(u,v)\in S} E(u,v) \tag{1}$$

where S={(u,v)|-R≤u,v≤R} is the *candidate set*, and R is the maximum search distance. In most cases, sum-of-absolute difference (SAD) between the target block and the reference block as given in (2) is used as the matching error because of its simplicity.

$$\Phi(u,v) = \sum_{i=0}^{k-1}\sum_{j=0}^{k-1}\left|F_C(x+i,y+j)-F_R(x+u+i,y+v+j)\right| \tag{2}$$

where k is the block size.

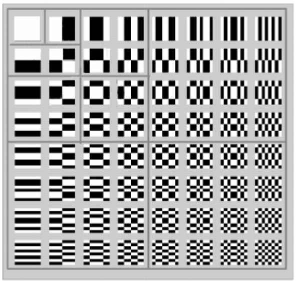


Fig. 1. BPs [1, 2] of 8×8 WHT

In this paper, we propose to perform block matching in WH domain and a *partial absolute distance* (PAD) $\Phi_p(u,v;q)$ is used as the matching error to reduce the computation requirement where q is the number of projections onto the WHT BPs. PAD may be regarded as an approximation of the SAD but requires significantly less computations. We shall show that block matching using PAD can find matching blocks of mean square error very close to that of SAD.

Suppose a $k\times k$ target block at (x,y) in current frame F_c is matched with a reference block of the same dimension at $(x+u,y+v)$ in its search area in reference frame F_R . The target block and the reference block are represented by vectors $\mathbf{b}_\mathrm{T}$ at (x,y) and $\mathbf{b}_\mathrm{R}$ at $(x+u,y+v)$ respectively in space $\Re^{k\times k}$. A difference vector $\mathbf{d}$ between $\mathbf{b}_\mathrm{T}$ and $\mathbf{b}_\mathrm{R}$ is defined as

$$\mathbf{d} = \mathbf{b}_\mathbf{T} - \mathbf{b}_\mathbf{R} \cdot \tag{3}$$

The SAD d between the reference block and the target block is shown in (4) where $\|.\|_p$ is the p-norm of a vector.

$$d = \|\mathbf{d}\|_1 = \|\mathbf{b}_\mathbf{T} - \mathbf{b}_\mathbf{R}\|_1 \tag{4}$$

Let S_q be a set of index (m,n), and each of them represents the number of horizontal and vertical zero-crossing of the first q BPs along the zigzag path. Projecting $\mathbf{b}_\mathrm{T}$ and $\mathbf{b}_\mathrm{R}$ onto BPs with indices in S_q, we get sets of $c_T(x,y;m,n;k)$ and $c_R(x+u,y+v;m,n;k)$ respectively. The $\Phi_p(u,v;q)$ between $\mathbf{b}_\mathrm{T}$ and $\mathbf{b}_\mathrm{R}$ is then defined by projecting $\mathbf{d}$ onto q WHT BPs as shown in (5).

If an additional WHT BP $\mathbf{h}$ is added into S_q, then PAD can be refined iteratively using (6) and becomes closer to SAD. Those reference blocks with PAD greater than a given threshold T_Φ will be rejected, and we search for the best match among the remaining candidates only.

$$\begin{aligned} d &\geq \Phi_p(u,v;q) \\ &= \sum_{(m,n)\in S_q} \frac{\left|\mathbf{h}_{(m,n)}^T \mathbf{d}\right|}{\left|\mathbf{h}_{(m,n)}\right|} = \sum_{(m,n)\in S_q} \frac{\left|\mathbf{h}_{(m,n)}^T \mathbf{b}_T - \mathbf{h}_{(m,n)}^T \mathbf{b}_R\right|}{\left|\mathbf{h}_{(m,n)}\right|} \\ &= \sum_{(m,n)\in S_q} \frac{\left|c_T(x,y;m,n;k) - c_R(x+u,y+v;m,n;k)\right|}{\left|\mathbf{h}_{(m,n)}\right|} \end{aligned} \tag{5}$$

$$\Phi_p(u,v;q+1) = \Phi_p(u,v;q) + \frac{\left|\mathbf{h}^T\mathbf{b}_T - \mathbf{h}^T\mathbf{b}_R\right|}{|\mathbf{h}|} \tag{6}$$

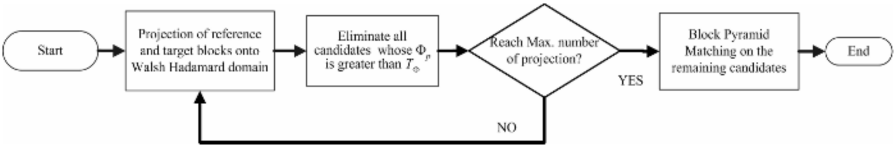

Start
Projection of reference and target blocks onto Walsh Hadamard domain
Eliminate all candidates whose Φ_p is greater than T_Φ
Reach Max. number of projection?
YES
NO
Block Pyramid Matching on the remaining candidates
End

Fig. 2. Flowchart of the proposed FWS system

The proposed motion estimation algorithm can perform fast block matching because of two reasons. Firstly, the low sequency order WHT BPs are highly probable to be parallel to the difference vector **d** so that the first few projections can acquire most of the distance between the targets block $\mathbf{b}_T$ and their candidates block $\mathbf{b}_R$. The second reason is the fast pruning algorithm of WHT, which computes the projections of the candidates in reference frames onto various BPs efficiently. We use a recursive structure of Walsh Hadamard tree [1, 2] in which the calculations applied to one candidate in reference frame or one BP projection are exploited when the projections of candidate or the projections onto another BP are computed.

The flowchart of the algorithm is shown in Fig. 2. To begin with, our algorithm computes PAD of each candidate in reference frame according to the corresponding WHT coefficients of the target blocks and reference blocks. If the PAD of a candidate is greater than a given threshold T_Φ, the location will be rejected. The remaining candidates in the reference frame are projected onto the higher sequency order WHT BPs. The PAD comparison repeats until a predefined number of projections is reached because the block matching in the WH domain is efficient only when the number of projections is small. We found that efficient block matching completely in WH domain is still possible by using a technique called pyramid block matching, which will be explained in the next section. In our implementation, only two projections are used to find the PAD. More projections require more computation but do not reduce MSE significantly.

The computation of PAD includes the transformation of frames, and the accumulation of absolute differences of WHT coefficients. Transforming reference and target frames requires about 8 operations, which include additions, subtractions, and absolute, per pixel for 2 projections 8×8 block [2, 3]. Total number of operations per pixel required to find PAD of the first and the second projections for one block is

$$N_{o,PAD} = \frac{1}{k^2}(2R+1)^2(2+3P_1) \tag{7}$$

where P_1 is the percentage of candidates remains after first projection.

3 Block Pyramid Matching

Hel-Or et al. suggest that the best matching position is the one with the minimum sum-of-squared distances (SSD) among the remaining candidates. However, the computation requirement of SSD is heavy, and we propose to use a block pyramid matching scheme to find a distance approximating the SAD such that computation can be reduced while not affecting the MSE performance much. In the first stage of BPM, each k×k block in reference frame and current frame is decomposed into four non-overlapping k/2×k/2 sub-blocks, and the projection of each k×k block onto $\mathbf{h}_{(0,0)}$

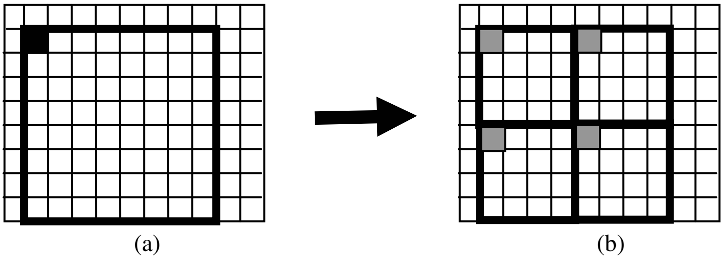


Fig. 3. Illustrate the locality relationship between (a) the WHT coefficient of the $k{\times}k$ block and (b) that of its corresponding $k/2{\times}k/2$ sub-blocks

(shaded box in Fig. 3a) is expressed as the sum of projections of the corresponding sub-blocks onto $\mathbf{h}_{(0,0)}$ (shaded boxes in Fig. 3b). Therefore, $\Phi_p(u,v;1)$ can be formulated as (8) when k=8.

$$\Phi_p(u,v;1) = \frac{|S_{T1} - S_{R1}|}{|\mathbf{h}_{(0,0)}|} \tag{8}$$

where

$$S_{T1} = \sum_{i=0,4}\sum_{j=0,4} c_T(x',y';\frac{k}{2})$$
$$S_{R1} = \sum_{i=0,4}\sum_{j=0,4} c_R(x'+u,y'+v;\frac{k}{2})$$
$$x' = x + i$$
$$y' = y + j$$

The relationship of the coefficient of blocks and their sub-blocks is illustrated in Fig. 3. We define the first level BPM estimation based on the projection onto $\mathbf{h}_{(0,0)}$ as $E_1(0,0)$, and it is shown in (9).

$$E_1(0,0) = \frac{\sum_{i=0,4}\sum_{j=0,4}\left|c_T(x',y';\frac{k}{2}) - c_R(x'+u,y'+v;\frac{k}{2})\right|}{|\mathbf{h}_{(0,0)}|} \tag{9}$$

Because of Triangular Inequality in (10),

$$\left|\sum_{j=1}^{M} a_j\right| \le \sum_{j=1}^{M} |a_j| \tag{10}$$

where $a_j \in \mathbf{R}$ and $M \in \mathbf{Z}^+$, $E_1(0,0)$ is closer to the SAD than $\Phi_p(u,v;1)$, but is still smaller than or equal to the SAD. In other words, $E_1(0,0)$ is a more accurate estimation of the SAD than $\Phi_p(u,v;1)$, i.e.

$$d \ge E_1(0,0) \ge \Phi_p(u,v;1) \tag{11}$$

It should be noted that the projection of $k/2{\times}k/2$ sub-blocks onto $\mathbf{h}_{(0,0)}$ are the intermediate data in the calculation of the WHT coefficient of $k{\times}k$ blocks using the recursive WH tree [1,2] . As a result, evaluating $E_1(0,0)$ requires much fewer computations than that of SAD, and contribute to the success of our fast block matching algorithm.

In the second stage of BPM, each $k/2\times k/2$ sub-block is further decomposed into four $k/4\times k/4$ sub-blocks. The projection of each $k/2\times k/2$ sub-block onto $\mathbf{h}_{(0,0)}$ (shaded box in Fig. 4a) can be expressed as the sum of four projections of the corresponding sub-blocks onto $\mathbf{h}_{(0,0)}$ (shaded boxes in Fig. 4b), which are available when we calculate the WHT coefficient of $k\times k$ blocks. In this stage, the $k\times k$ block is divided into sixteen $k/4 \times k/4$ sub-blocks. The first level BPM estimation $E_1(0,0)$ can then be expressed as (12).

$$E_1(0,0)=\sum_{i=0,4}\sum_{j=0,4}\left|S_{T2}-S_{R2}\right| \tag{12}$$

where

$$S_{T2}=\sum_{m=0,2}\sum_{n=0,2}c_T(x'',y'';\frac{k}{4})$$

$$S_{R2}=\sum_{m=0,2}\sum_{n=0,2}c_R(x''+u,y''+v;\frac{k}{4})$$

$x''=x+i+m,$ and $y''=y+j+n$.

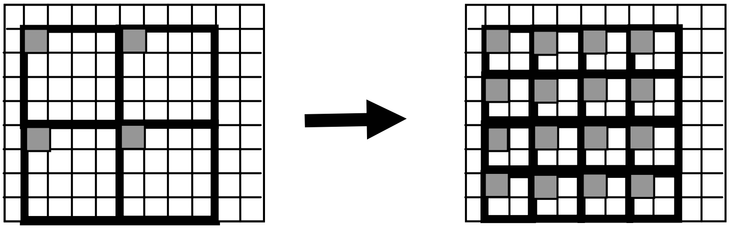
Similar to the first level BPM estimation, we define the second level BPM estimation $E_2(0,0)$ based on the projection of $\mathbf{h}_{(0,0)}$ as

$$E_2(0,0)=\frac{\sum\limits_{i=0,4}\sum\limits_{j=0,4}\sum\limits_{m=0,2}\sum\limits_{n=0,2}\left|c_T(x'',y'';\frac{k}{4})-c_R(x''+u,y''+v;\frac{k}{4})\right|}{\left|\mathbf{h}_{(0,0)}\right|}. \tag{13}$$

Because of Triangular Inequality, $E_2(0,0)$ is more accurate to approximate d than $E_1(0,0)$ as shown in (14). Theoretically blocks can be decomposed further until the block size becomes one. In that case the BPM estimation becomes the SAD itself.

$$d\geq E_2(0,0)\geq E_1(0,0)\geq \Phi_p(u,v;1) \tag{14}$$

In our previous discussion, we concern with the BPM based on the projection onto $\mathbf{h}_{(0,0)}$ only. It can be shown that $E_1(0,0)$ is also the first level BPM based on the



(a) (b)

Fig. 4. Illustrate the locality relationship between (a) the WHT coefficients of the $k/2 \times k/2$ block and (b) that of its corresponding $k/4 \times k/4$ sub-blocks

projection onto $\mathbf{h}_{(a,b)}$ where $0 \le a,b \le 1$. Similarly, $E_2(0,0)$ is the second level BPM based on the projection onto $\mathbf{h}_{(c,d)}$ where $0 \le c,d \le 3$, i.e.

$$E_1(a,b) = E_1(0,0) \tag{15}$$

$$E_2(c,d) = E_2(0,0) \tag{16}$$

where $0 \le a,b \le 1$ and $0 \le c,d \le 3$.

Therefore, when we compare $E_1(0,0)$ or $E_2(0,0)$ of the target block and the reference blocks, we have already compared their $E_1(a,b)$ or $E_2(c,d)$ where $0 \le a,b \le 1$ and $0 \le c,d \le 3$ respectively. In other words, we have used the information from the projections onto higher sequency order WHT BPs to get more precise similarity evaluation when we compare the corresponding $E_1(0,0)$ and $E_2(0,0)$ of the target block and the reference blocks.

In the proposed algorithm, after rejecting candidates in reference frame using PAD, the best K_1 % of the remaining candidates, i.e. those with the least PAD, will go through the first level BPM in which the $E_1(0,0)$ difference between the target block and the remaining candidates are computed. Then, the best K_2% candidates after first level BPM will be further examined by evaluating their $E_2(0,0)$ difference. The candidate with smallest $E_2(0,0)$ difference between the target block is elected as the best match of the target block, and will be regarded as the location pointed by the corresponding motion vector. Assuming the maximum allowed candidates are used for first and second stage of BPM, the number of operations required to find the best match per pixel is

$$N_{o,BPM} = \frac{1}{k^2}(2R+1)^2\left[K_1\left(3\left(\frac{k}{4}\right)^2 - 1\right) + K_2\left(3\left(\frac{k}{2}\right)^2 - 1\right)\right]. \tag{17}$$

4 Experimental Results

We applied the FWS to 80 frames of three standard sequences: Football, Foreman, and Stefan with k=8 and R=16. For each candidate in reference frame, the maximum number of projections allowed is two and the remaining candidates will go through BPM. The threshold T_Φ is 10. For BPM, K_1=10% and K_2=5%.

4.1 Computation Requirement

Finding the SAD of a $k \times k$ block requires k^2 subtractions, k^2 absolute operations, and k^2-1 additions; therefore, the total number of operations is $3k^2-1$. In FSBM, the number of candidate for each block is $(2R+1)^2$, and each frame with A number of pixels has number of blocks A/k^2. Then the total number of operations $N_{o,FS}$ required per pixel is

$$N_{o,FS} = \frac{1}{k^2}(2R+1)^2(3k^2-1). \tag{18}$$

With k=8 and R=16, $N_{o,FS}$ = 3245. On the other hand, TSS has only $8\log_2 R+1$ candidates for each block, therefore, the total number of operations per pixel, $N_{o,TSS}$, becomes

$$N_{o,TSS} = \frac{1}{k^2}(8\log_2 R+1)(3k^2-1). \tag{19}$$

In our experiment, $N_{o,TSS} = 98$. Since DS has no fixed number of search points, its computation varies for different videos. According to [7], DS has a computation of about 80% of TSS. Similar to DS, the computation of FWS also varies for different videos since the numbers of remaining candidates after each projection are different. Experimental results show that around 40% of candidates remain after first projection, and 25% remains after second projection. Table 1 shows the total number of addition, subtraction and absolute operations required per pixel for FS, TSS, and FWS. The computation of the proposed FWS includes WHT of frames, PAD, and BPM computations. FWS usually requires about 20% more computation than TSS. Because the intermediate data in the recursive WH tree are reused, more memory is needed compared to FS and TSS.

4.2 MSE Performance

Experimental results show that two projections are enough and additional projections do not reduce MSE significantly, but will increase the computation time. About 75% of candidates will be eliminated after two projections. Table 2 shows the average MSE over 80 frames of the three sequences using different algorithms and Fig. 5 shows the MSE of the each frame. The performance of FWS in terms of MSE is very close to FS, but the computation required is only a little bit more than TSS. TSS and DS, while much faster than FS, produce MSE which are significantly larger than FS and FWS.

Replacing SAD by BPM after two projections can significantly reduce computations. The resultants MSE, however, are not affected much. Table 3 shows the increase in MSE when SAD is replaced by BPM. On average, the MSE is increased by merely 5%.

Table 1. Operations per pixel needed for different search methods

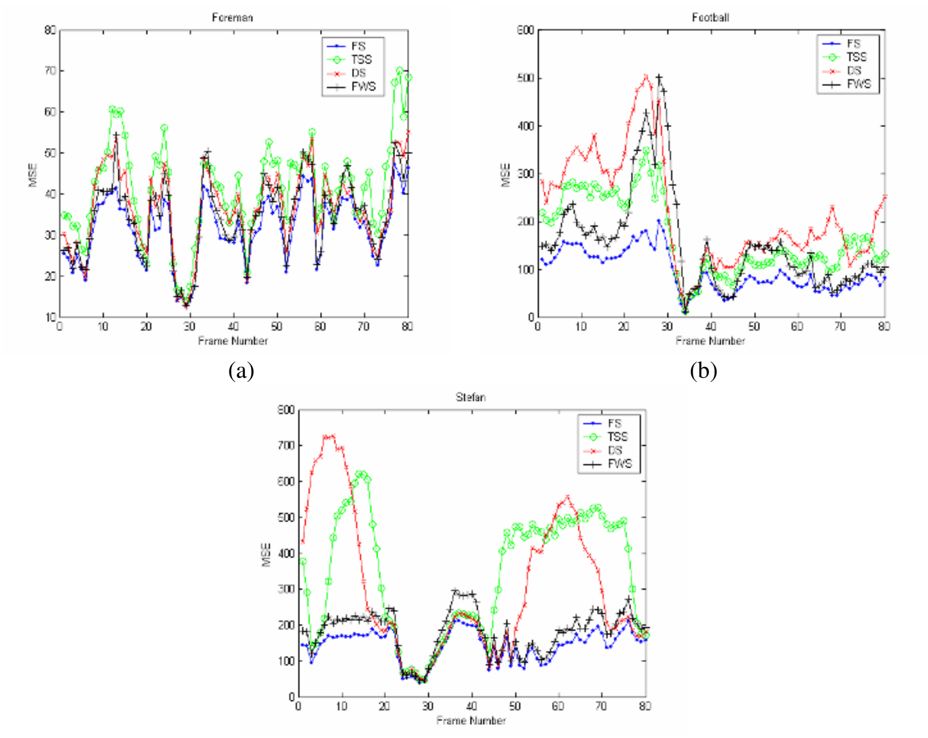
Sequence	FS	TSS	FWS
Foreman	3245	98	118
Football	3245	98	126
Stefan	3245	98	123

Table 2. Average mean-squared-error of 80 frames

Sequence	FS	TSS	DS	FWS
Foreman	31.5	41.1	36.4	34.5
Football	94.4	167.0	220.0	155.4
Stefan	142.5	341.1	308.0	181.3

Table 3. MSE comparison of SAD and BPM

Sequence	MSE		
	SAD	BPM	MSE
Foreman	32.8	34.5	+5.4%
Football	148.8	155.4	+4.4%
Stefan	169.2	181.3	+7.1%



(a) (b)

(c)

Fig. 5. MSE plots for sequence (a) Foreman (b) Football (c) Stefan

5 Conclusions

A fast block matching method, FWS, which is based on a pattern matching algorithm in Walsh Hadamard domain, is proposed in this paper. The computation requirement is similar to the three-step-search, but the accuracy is comparable with the full-search method. Efficient projection scheme is utilized for fast WHT. Furthermore, we exploit the intermediate results in WHT calculation to reject candidate blocks that are unlikely to be a good match. Both measures significantly reduce computations in the block matching process. Experimental results show that the performance of FWS in terms of MSE is very close to that produced by full search algorithm.

References

[1] Y. Hel-Or; H. Hel-Or; “Real time pattern matching using projection kernels”, Proc. of Ninth IEEE International Conference on Computer Vision, Vol. 1, pp. 1486 – 1493, Oct. 2003.

[2] Y. Hel-Or; H. Hel-Or; “Real time pattern matching using projection kernels”, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 27, No. 9, Sept 2005.

[3] W. K. Cham; R. J. Clarke; "Application of the principle of dyadic symmetry to the generation of orthogonal transforms", IEE Proc. F, Commun., Radar & Signal Process., Vol. 133, no.3, pp.264-270, June 1986.

[4] T. Koga; K. Iinuma; A. Hirano; Y. Iijima; T. Ishiguro; "Motion compensated interframe coding for video conferencing," in Proc. Nat. Telecommun. Conf., New Orleans, LA, Nov. 29-Dec. 3 1981, pp. G5.3.1-5.3.5.

[5] Lai-Man Po; Wing-Chung Ma; "A novel four-step search algorithm for fast block motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, No. 3, June 1996, pp. 313 - 317.

[6] Reoxiang Li; Bing Zeng; Liou, M.L.; "A new three-step search algorithm for block motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 4, No. 4, Aug. 1994, pp.438 - 442.

[7] Shan Zhu; Kai-Kuang Ma; "A new diamond search algorithm for fast block-matching motion estimation," IEEE Transactions on Image Processing, Vol. 9, No. 2, Feb. 2000, pp. 287 - 290.

[8] T. Wiegand; G. J. Sullivan; G. Bjontegaard; A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol, Vol. 13, pp. 560-576, July 2003.

Skin Detection by Near Infrared Multi-band for Driver Support System

Yasuhiro Suzuki[1], Kazuhiko Yamamoto[1], Kunihito Kato[1],
Michinori Andoh[2], and Shinichi Kojima[2]

[1] Faculty of Engineering, Gifu University 1-1 Yanagito, Gifu 501-1193, Japan
suzukiy@yam.info.gifu-u.ac.jp,
{yamamoto, kkato}@info.gifu-u.ac.jp
[2] TOYOTA Central R&D Labs., Inc. Nagakute, Aichi 480-1192, Japan
{andoh, skojima}@mosk.tytlabs.co.jp

Abstract. Many active safety technologies for the driver support system are developed. Most of the traffic accidents are caused by driver's inattentive or drowsy. We are developing a driver support system that protects from traffic accidents by these causes. Our purpose is to detect the driver's face region. A lot of face detection methods are proposed, but there is not a technique addressing every environment inside the car. In this paper, we propose a skin detection method by the unique reflection characteristics of the materials. We developed the skin detection system, and confirmed the effectiveness by the evaluation experiment.

1 Introduction

In recent years ITS technologies are developed and used practically in various fields [1]. A study of the Advanced Safety Vehicle (ASV) becomes popular. A lot of researches of the system to improve safety, operability and convenience by a camera put on outside and inside of the car are proposed [2][3][4].

We are developing a system that supports the driver by estimating state of the driver. At first, to estimate state of the driver, a face region of the driver has to be detected from the input image. A lot of face detection methods are proposed [5][6][7], but these does not yet address all environment in the car. Usually, most methods use skin color segmentation [8] and template matching [9]. There methods have many problems which are influence of lighting conditions, matching error, the calculation cost and etc..

We had proposed the face region detection system by switching ON/OFF of near infrared (IR) light and using a band-pass filter [10]. However, it is difficult to distinguish the face and the headrest, because the region which is illuminated IR was defined face region.

Therefore we paid attention to the reflectance characteristics of individual material in the near IR spectrum. Considering about the face, there is much difference in the reflection characteristic of the material between hair and skin. Therefore, we can detect skin region if the feature of the reflection characteristic is extracted. In addition, we can irradiate strong IR light enough to detect, because near IR light is

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 722 – 731, 2006.

invisible to the human eye. The purpose of our research is to extract the material of skin by the near IR spectrum multi-band directly.

2 System Configuration

In this research, the camera is installed on front of the meter as shown in Fig.1, and we studied basic experiment and verification in the room. Fig.2 shows the overview of this system. CCD camera and the near IR illuminator are set up for driver's face. The image is acquired by using near IR-LED for the irradiation equipment by installing near-IR penetration filter (IR-Pass-Filter) that penetrates only the near-IR light in the camera.

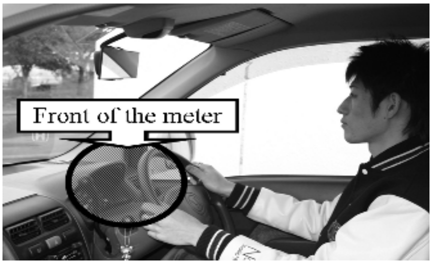


Fig. 1. Camera position

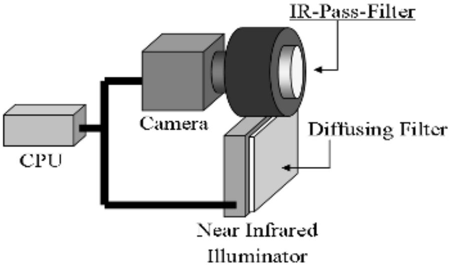


Fig. 2. System configuration

2.1 Effectiveness of Near IR Light

This system illuminates light to the driver's face and takes the image. This chapter is considered the selection of the band wavelength for the irradiation equipment.

In the case of using visible light, skin region is detected by using the skin color segmentation. In the nighttime, it is necessary to illuminate strong light to the driver in the dark car. This light will be stress dazzling, and of cause it will be an obstacle to driving. Therefore, it is difficult to use only a visible optical band.

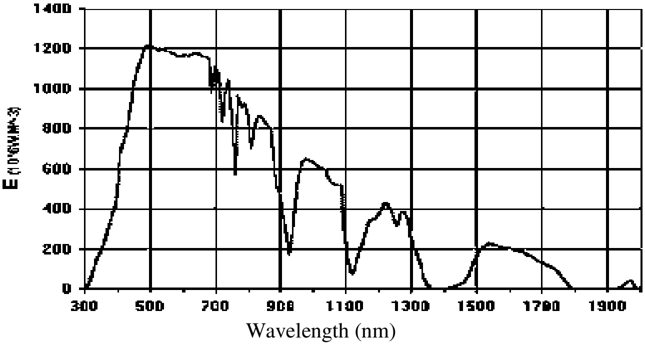


Fig. 3. Solar radiation spectrum at the earth's surface

On the other hand, in the case of using IR light, this light does not give the stress for driver, because it is invisible. Therefore, enough level of light can be illuminated even if it is in the night. Fig.3 shows spectrum distribution of the sun light. In Fig.3, IR light intensity is less than visible light intensity.

Therefore, by using a filter that penetrates only IR light (IR-Pass-Filter) and a specific band wave length (BP-IR-Filter), it is able to construct the illuminant system that is not influenced by ambient light such as the sun light. Consequently, the effectiveness of the near IR light can be shown.

3 Detection Method of Skin Region

3.1 Reflection Characteristic of Skin and Hair

Fig.4 shows the reflection characteristic of the skin and the hair [11]. It is generally said that the visible optical area is 380~760nm, and the near-IR radiation area is 800~2400nm.

We defined that the region 800~1400nm is the lower band and the region 1400~2400nm is the upper band in the near IR region. It is shown in the Fig.4 that the skin the lower near IR band more than the hair. On the other hand, the hair reflects the upper near IR band more than the skin. Therefore, the skin can be detected by taking subtraction between the irradiation image of a lower band and the irradiation image of the upper band.

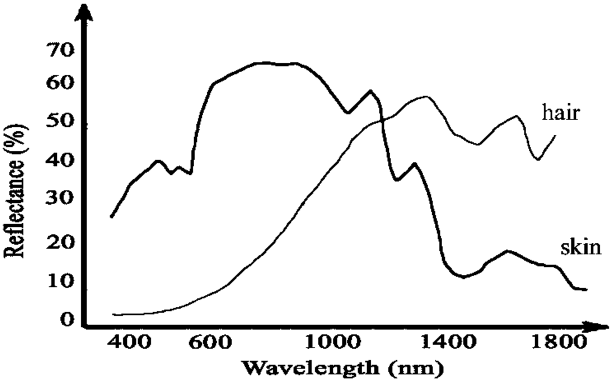


Fig. 4. Reflectance characteristics of skin and hair

Dowdall developed a multi-band system by three bands of the visible band, the lower band and the upper band, and he proposed a technique for detecting the camouflaged person by the method of the skin region detection [12]. He used three cameras corresponding to those bands. Therefore, three cameras for each light source and three wavelengths were needed.

The purpose of this paper is to propose a technique for extracting the material of the skin that composes the face by only one camera and two wavelengths. We propose a face extraction technique based on the characteristic of the materials.

3.2 Near-IR Multi-band Illuminant

The camera which has dynamic sensitivity range from the lower band to the upper band is very expensive. But, the camera of this system is required to be inexpensive. We use the CCD camera with sensitivity range 400~1000nm which is XC-EI50 made by SONY.

The available sensitivity band is limited as had described in the preceding chapter. Therefore, we choose 870nm and 970nm as the spectrum of multi-band. Table 1 shows the reflection characteristic of the skin and the hair between these two wavelengths.

Table 1. Difference of reflection characteristic of skin and hair

Skin	The reflectivity of 870nm is higher
Hair	The reflectivity of 970nm is higher

The irradiation equipment is made by using two kinds of IR-LED with different output wave length [13]. Fig.5 shows the arrangement pattern of IR-LED of two wavelengths. The diffusion filter which is set in front of the illuminator was resolved the inhomogeneous irradiation of two wavelengths.

The output of IR-LED is set at the level that there is no influence on the human body according to JIS (Japanese Industrial Standards).

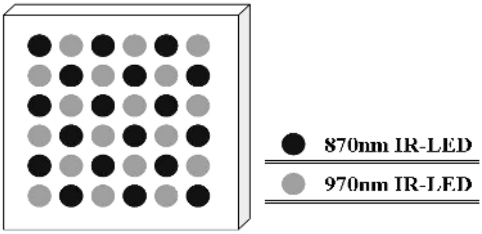


Fig. 5. LED arrangement pattern

4 Fundamental Experiment in Indoor Environment

The preceding chapter showed a method to distinguish each material by detecting the reflection characteristics of skin and hair. In this chapter, we considered the effectiveness of this method in indoor environment.

4.1 Skin Detection Method

We can distinguish a material of the skin or the hair by only subtraction of two images which are taken with irradiating two several frequencies.

At first, an image irradiated 870nm is taken, and then an image irradiated 970nm is taken by same method. Fig.6(a) is an image irradiated 870nm, and Fig.6(b) is an image irradiated 970nm. These images are defined I_{870} and I_{970}. Let $f_{870}(i,j)$ and $f_{970}(i,j)$ be pixels of images I_{870} and I_{970}, the subtraction value $f_s(i, j)$ is expressed eq.(1).

$$\begin{aligned} f_s(i,\ j) &= f_{870}(i, j) - f_{970}(i, j)\ . \\ \because\ I_{870} - I_{970} &= f_s(i, j)\ . \end{aligned} \tag{1}$$

In a wavelength of 870nm, the skin is the material that the reflection rate is high. Therefore, the difference value $f_s(i, j)$ becomes a positive value. On the other hand, the difference value $f_s(i, j)$ of hair becomes a negative value. The background where light does not reach is removed by subtraction.

The subtraction image I_s is expressed as follows. n is normalization value.

$$I_s = f_s(i,\ j) \times n\ . \tag{2}$$

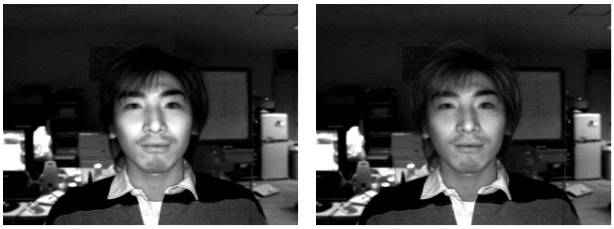

(a) 870nm (b) 970nm

Fig. 6. Irradiation image each wavelength

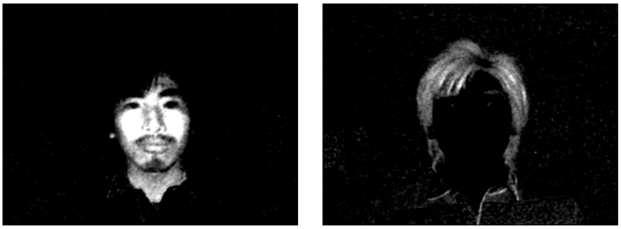

(a) Positive value (a) Binarization

Fig. 7. Subtraction result

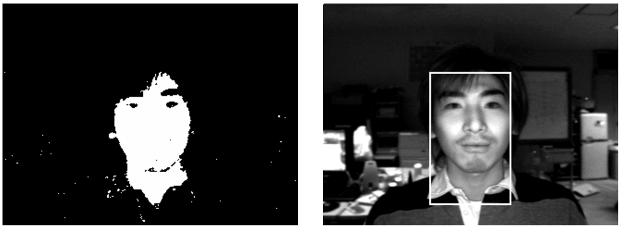

(a) Binarization (b) Skin region

Fig. 8. Skin region detection

Fig.7(a) shows region of positive difference values, and the skin region is detected. Fig.7(b) shows region of negative difference values, and also the hair region is detected.

It is easy to detect the skin region or the others. To detect the skin region, at first, the image which has a positive value is binarized (as shown Fig.8(a)). The image of Fig.8(b) shows the hugest region of the binarized image by the labeling operation. We see from Fig.8(b) that the search result is skin region.

4.2 Normalization of Distance

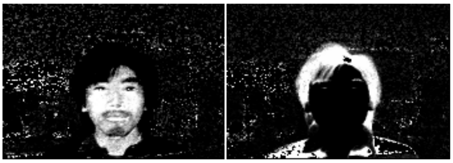
The unique reflection characteristic of the material is extracted by simple subtraction (eq.(1)). However, the subtraction value is influenced by the attenuation of light when the distance for the subject changed. A face position is not fixed at all in a driving person. Therefore, the subtraction value has to be normalized in distance. We normalize the subtraction value to distance. The normalized value $f_n(i, j)$ is expressed in eq(3).

$$f_n(i,j) = f_{870}(i,j) - f_{970}(i,j) \,/\, f_{870}(i,j) + f_{970}(i,j) \qquad [-1 < f_n(i,j) < 1] \tag{3}$$

Since attenuation of light of two lengths are linear level corresponding to distance, the normalized value will be equal even if any distance. The subtraction image I_n is expressed as follows. m is normalization value.

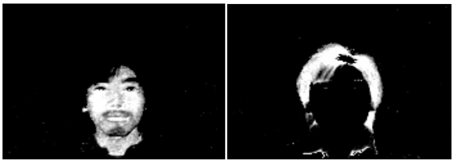
$$I_n = f_n(i,j) \times m\,. \tag{4}$$

Fig.9(a) shows region of positive difference values, and Fig.9(b) shows region of negative difference values.



(a) Positive value (b) Negative value

Fig. 9. Subtraction result by eq.(4)



(a) Positive value (b) Negative value

Fig. 10. Normalization image

Noise is occurred in two images as shown in Fig.9. This cause is that it is influenced a value of the denominator greater than the molecule in eq.(3).

However, as for the molecule and the denominator of eq.(3), the attenuation of light is stored in the both values. Therefore we gave the condition types as follows.

$$\max(f_{870} - f_{970},\, p\,)\,. \tag{5}$$

$$\max(f_{870} + f_{970},\, q\,)\,. \tag{6}$$

When the subtraction value is smaller than ***p*** of error, the region of the value is considered a background (as shown in eq.(5)). In the same reason, when the sum value is smaller than ***q*** of the attenuation of light, the region of the value is considered background (as shown in eq.(6)). The subtraction image is taken from two input images of Fig.6 by adding the condition types. Fig.10(a) shows region of positive difference values, and Fig.10(b) shows region of negative difference values.

In Fig.10, influence of background is decreased, and the reflection characteristics are extracted then Fig.9. Furthermore, influence of shape of a face is decreased then Fig.7(a). Therefore, the reflection characteristic of skin is clearly appeared.

4.3 Comparative Experiments for Distance

We conducted comparative experiments about the technique of normalization by eq.(3). We show an experiment procedure in the following.

We experimented in the room, and the illumination environment is a general illumination. The subject is the same person. We took the images of two lengths in distance of 50cm~80cm with an interval 10cm from the system. We calculated each subtraction values by eq.(1) and eq.(3). For eq.(3), parameters were set as $\boldsymbol{p}$=3, $\boldsymbol{q}$=10 in eq.(5) and eq.(6). We took average of the positive difference value as skin region. The graph in Fig.11 shows the subtraction values in each distance. The both difference value are normalized in $-1<\boldsymbol{N}<1$.

From Fig.11, the values of eq.(3) are stably in any distances. Consequently, the subtraction value in material distinction is stable, besides setting of the threshold becomes easier.

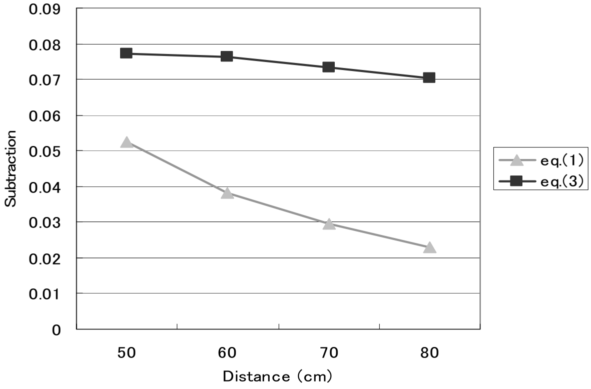


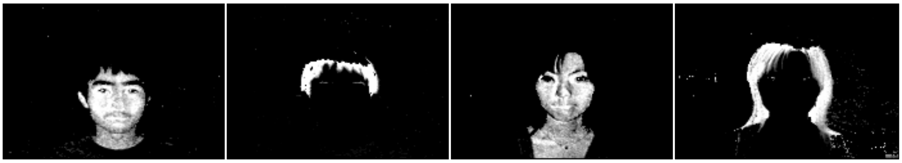
Fig. 11. Average value of the difference of two methods

4.4 Experiments of Skin Region Detection

We experimented skin region detection by using eq.(3). The experimental condition is same as previous experiment. The subjects were twenty four men and four women sum total twenty eight, and nobody wear glasses. Results were checked the success or failures from manually check. In the case of detecting skin region, it was evaluated success.

Fig.12 shows examples of the subtraction image as a result of experiment. Fig.12 takes a look at some examples of man and woman. Fig.12(a) shows region of positive difference values, and Fig.12(b) shows region of negative difference values.

From the result, 100% (28/28) detection rate was given in indoor environment.



(a) Positive value (b) Negative value (a) Positive value (b) Negative value

Man Woman

Fig. 12. Examples of experimental result

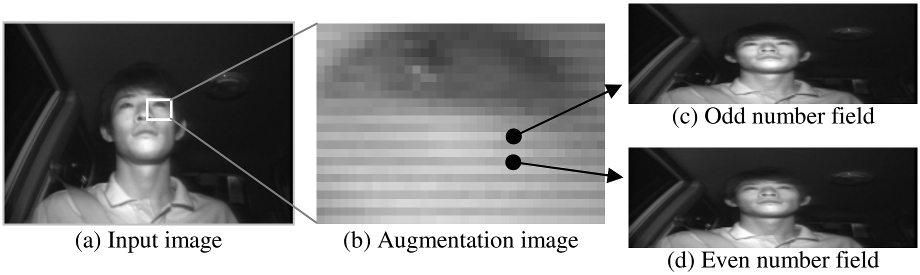
5 Experiments in the In-Vehicle Environment

In the preceding chapter, it was suggested the skin detection method and observed the effectiveness of our method by the basic experiment in the room. Therefore, we experimented it on the car in the night when skin color detection is impossible.

5.1 Experiment Method

We experimented on the in-vehicle environment to confirm the effectiveness of our method in the night. The experimental time is the approx. 20 minutes ride in the urban area from the suburbs without a streetlight. The experiment has done at 19:00 after sunset. A subject did not wear glasses. Results were checked the success or not from manually check. In the case of detecting skin region, it was evaluated success.

The irradiation method is devised because a subject did not always stop in the driving. When we got an image of NTSC, the illuminator emitted light of 870nm at the time of odd number field in one frame, and emitted light of 970nm at the time of even number field. Fig.13 shows a provided image. This method can get two wavelength images from one NTSC image. Therefore, real-time processing of 30 frames second is enabled without reducing the frame rate. It was able to take in influence from rolling of the head of a driver.



(a) Input image (b) Augmentation image (c) Odd number field (d) Even number field

Fig. 13. Input image in the experiment

5.2 Experimental Result of Night Driving

As well as shine of street lights of a town area, light of headlights of oncoming cars illuminated the face of a driver. There was not the influence of those light. Because of the system was set the filter transmitting only by near IR light in front of the camera, and illuminated strong near IR spectrum than ambient lights.

As a consequence of this experiment, face region detection rate was 100% (36000/36000 [frames]) for 20 minutes from the suburbs to the urban area.

6 Conclusion

We proposed that detection method of the skin region by focusing on the unique reflection characteristic in materials of skin and hair. By using multiple near IR bands, the skin region is easily detected only by subtraction of the two images. We confirmed the effectiveness in indoor environment by the evaluation experiment and stability by the distance normalization to the subject. Furthermore, it was shown that this method is effective by the in-vehicle experiments.

Near IR light is invisible to the human eye the system cannot be stress dazzling in the night. Moreover, this method can extract directly material of human skin. Moreover, the system cost and the calculation cost are a few because the system and the algorithm are simple.

For future work, we are going to experiment in-vehicle environment, and develop this method's application for detection inattentive and drowsy driving.

References

1. Y.Natsume: "Improvement of Car Traffic Safety by ITS", Trans. EISS, pp.361-362, 1998 (in Japanese)
2. Y.Ninomiya, A.Takahashi, and M.Ohta: "Lane Recognition System Based on the High-Speed Pattern Matching Method", Trans. IEICE, Vo1.J86-D2, No.5, pp.625-632, 2003 (in Japanese)
3. J.Fukuda, K.Adachi, M.Nishida, and E.Akutsu: "Development of Driver's Drowsiness Detection Technology", TOYOTA Technical Review Vol.45 No.1, pp.34-40, 1995
4. Q.Ji, Z.Zhu, and P.Lan, "Real Time Non-intrusive Monitoring and Prediction of Driver Fatigue", to appear in IEEE Transactions on Vehicular Technology, 2004
5. S.Akamatsu: "Computer Recognition of Human Face —A Survey—", Trans. IEICE, Vo1.J80-D2, No.8, PP.2031-2046, 1997 (in Japanese)
6. O.Yamaguchi and K.Fukui: ""Smartface"—A Robust Face Recognition System under Varying Facial Pose and Expression", Trans. IEICE, Vol.J84-D2, No.6, pp.1045-1052, 2001 (in Japanese)
7. K.Hotta, T.Kurita, S.Umeyama, and T.Mishima, "Face Matching through Information Theoretical Attention Points and Its Applications to Face Detection and Classification," Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp.34-39, 2000
8. H.Wu, Q.Chen, and M.Yachida, "Face detection from color images using a fuzzy pattern matching method", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.21, No.6, pp.557-563, 1999

9. Y.Ishii, H.Hongo, K.Yamamoto, and Y.Niwa: “Discussion on facial features and the regions for face detection”, Trans. J-FACE, Vol.3, No.1, pp.33-41, 2003 (in Japanese)
10. I.Takai, K.Yamamoto, K.Kato, K.Yamada, M.Andoh: "Detection of the Face and Eye Region For Drivers’ Support System", Proc. QCAV2003, pp.375-380, 2003.5
11. J.Dowdall, I.Pav1idis, and G.Bebis: “Face Detection in the Near-IR Spectrum”, Proc. of Image and Vision Computing, Vo1.21, pp.565-578, 2003
12. I.Pav1idis and P.Symosek: “The Imaging Issue in an Automatic Face/Disguise Detection System”, Proc. of IEEE Workshop on Computer Vision beyond the Visib1e Spectrum: Methods and App1ications, PP.15-24, 2000
13. Y.Suzuki, K.Yamamoto, K.Kato, M.Andoh, and S.Kojima: "Human Detection by Using near Infrared Multi-Band for the Driver Support”, Proc. of Chinou Mechatronics Workshop, pp.224-229, 2005 (in Japanese)

Extracting Surface Representations from Rim Curves

Hai Chen[1], Kwan-Yee K. Wong[2], Chen Liang[2], and Yue Chen[1]

[1] College of Software Technology, Zhejiang University,
Hangzhou, Zhejiang, China 310027
chenhai@gmail.com
chenyue@cs.zju.edu.cn

[2] Department of Computer Science, The University of Hong Kong,
Pokfulam Road, Hong Kong
{kykwong, cliang}@cs.hku.hk

Abstract. In this paper, we design and implement a novel method for constructing a mixed triangle/quadrangle mesh from the 3D space curves (rims) estimated from the profiles of an object in an image sequence without knowing the original 3D topology of the object. To this aim, a contour data structure for representing visual hull, which is different from that for CT/MRI, is introduced. In this paper, we (1) solve the "branching structure" problem by introducing some additional "directed edge", and (2) extract a triangle/quadrangle closed mesh from the contour structure with an algorithm based on dynamic programming. Both theoretical demonstration and real world results show that our proposed method has sufficient robustness with respect to the complex topology of the object, and the extracted mesh is of high quality.

1 Introduction

In 3D model reconstruction from image sequences, silhouettes are often a reliable and obvious feature that can be extracted from the images easily. With the knowledge of the epipolar geometry, which governs the relative positions and orientations between cameras, it is possible to recover the 3D space curves lying on the surface of the object that are projected onto the images as the silhouettes. In the literature, such 3D space curves are known as contour generators or rims.

Various techniques[1, 2, 3] have been developed for estimating rims from silhouettes. The rims so obtained carry not only 3D positional information of the space curves, but also surface information like the surface normal. It is desirable to extract a surface representation of the object from the rims. However, this is not a trivial task as the rims cannot be recovered perfectly. Very often, the rims recovered may be discontinuous due to self-occlusion. Besides, they may intersect with each other at frontier points (see [1] for details). The points forming the rims are also not evenly spaced. Hence, direct triangulation of the points on the rims using existing algorithms often cannot produce satisfactory results.

In this paper, a novel method for constructing a surface mesh from the rims estimated from the silhouettes is introduced. Instead of triangulating the rims points directly, the rims are first re-sampled by evenly spaced parallel slicing planes. The sample

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 732–741, 2006.

points on each slicing plane then forms the 2D cross-section contours of the object. The problem of forming a surface mesh is then converted into the problem of joining these cross-section contours on adjacent layers. Unlike cross-section data commonly seen in MRI/CT related research, we observe that the contours recovered from silhouette data do not always overlap with each other. This is the well-known problem of ”branching structure” in MRI/CT visualization. A method based on “directed edge” is introduced here to solve this problem. By exploiting the dynamic programming (DP) techniques, it is shown that the mesh ‘belt’ can be reconstructed from two adjacent cross-section contours with a low computation complexity, which is linear to the product of the numbers of the mesh vertices on the two contours. This technique also allows us to build a high quality mesh in terms of surface smoothness.

Another contribution of this paper is that the final surface can be presented in the form of a mixed triangle/quadrangle mesh, which can be rendered more efficiently than the pure triangle mesh. The mixed triangle/quadrangle mesh has only been used in mesh subdivision and mesh edit, and it is the first time that it is extracted directly from image data.

The latter of this section gives the overview of related works. Section 2 presents the theoretical background of mesh. Section 3 describes the algorithms and implementations for extracting a mesh surface from the rims. Experimental results from real models will be given in Section 4, and Section 5 concludes this paper.

1.1 Related Works

In the literature, there have been some related researches that attempt to extract a mesh from the rims of an object. In [2], the triangular mesh is extracted directly based on the relationship between neighboring rims with minimal computation complexity. However, such a relationship between rims will not hold if the rims are fragmentary, and this happens quite often for complex shapes with non-zero genus. Note that this approach makes no guarantee on the quality nor the obturation of the outcome mesh.

Since the connectivity information is implied for points recovered along the rims, such points can be reformed into another data structure which makes the surface extraction easier. In this paper, the rims are re-sampled into cross-section contours. There are numerous researches on the contour data, but most of them are based on contour data derived from CT/MRI. In [4], Cong and Parvin recovered a surface from planar sectional contours based on the “Equal Importance Criterion” which suggests that every point in the region contributes equally to the reconstruction process. This algorithm derives the iso-surface constructed by PDE and the primitive representations by Voronoi Diagram transformation. However, this approach is of very high complexity and not suitable for the surface reconstruction of the visual hull.

There are also attempts made to extract a surface from contours for visual hull reconstruction. In [5], Boissonnat exploited Delaunay’s triangulation, a method commonly used in reconstructing 3D surface from unorganized points, to extract a surface from the planar contour structure. The algorithm is robust but the resulting surface is not obturated. Besides, the algorithm produces low quality mesh near branching structures on the surface.

2 Theoretical Background

2.1 Quality of Mesh Elements

A mesh with high quality not only can faithfully capture the true topology of a complex 3D object, but also can be rendered efficiently. According to [6], a high quality visual hull surface should be a compact, connected, orientable, two dimensional manifold, and with or without boundary.

Let us first define quantitatively a measurement of the quality for the mesh elements. Traditionally, the quality of a mesh triangle is measured by the smallest internal angle, and the quality of the triangle is said to increase with its smallest internal angle [7]. Since in this paper, our output mesh will be a mixed triangle/quadrangle mesh (to be introduced in next subsection), the internal angle measurement cannot be applied. Here, a distance measurement similar to that used in [8] and [9] is used to measure the quality of triangles and quadrangles.

During the extraction from cross-section contours, every mesh elements are formed from points on the contours lying on adjacent layers, and are either a triangle or a quadrangle. A triangle consists of a vertex from the contour polygon on one layer and an edge of a contour polygon on another layer. By projecting the vertex onto the other contour plane, the ***Error of Triangle*** is defined as the squared distance between the projection (p_i) and the center (c_j) of the edge (see (1) and (2)).

A quadrangle consists of an edge of the contour polygons lying on one layer and an edge of the contour polygons lying on the other layer. By projecting the center of one edge onto the other contour plane, the ***Error of Quadrangle*** is the defined as twice the squared distance between the projection (c_i) and the center (c_j) the other edge (see (3)).

2.2 Mixed Triangle/Quadrangle Mesh

A mesh formed using the triangle scheme can retain sharp features more faithfully, while that formed using a quadrangle scheme is more suitable for representing smooth surfaces. Triangle meshes generate poor limiting surface when using quadrangle-only scheme, while quadrangle meshes behave poorly with triangle-only scheme. To increase flexibility, both triangle and quadrangle schemes are needed in modeling real world data. Recently, Stam and Loop [9] introduced a new subdivision operator that unifies mixed triangular and quadrilateral subdivision schemes on C^1 surfaces. Latter, Schaefer and Warren [8] proved that mixed triangle/quadrangle scheme mesh could be used in C^2 surfaces. Here, in this paper, the mixed triangle/quadrangle meshes on C^2 surfaces are extracted from the contour data structure directly.

3 Surface Extraction from Rims

In this work, the cross-section contours are first formed from the rim fragments estimated from the silhouettes. A dynamic programming based method is then introduced to produce a high quality triangle/quadrangle mesh from these cross-section contours. We will explain the algorithm and our implementation in detail.

3.1 Contour Data from Rims

To extracting a high quality mesh, the rims are first transformed into a more efficient data structure bearing the topological information observed from the silhouettes. A cross-section contour data structure is adopted here for representing the surface of the visual hull.

We adopt the method introduced in [1] to recover the rims from the silhouettes in an image sequence. These rims are then re-sampled into cross-section contours by parallel slicing planes. The normal of the slicing planes are chosen to be the direction parallel to the longest shaft of the original object. To recover the contour structure for complex models, we back-project the points onto the extracted silhouettes and regroup points into one or more contours on the same sliced plane. After regrouping, points on each cross-section form one or several planar polygons (contour polygons) which correctly capture the topology observed from the silhouettes.

As mentioned earlier, the recovered rims are inevitably fragmentary. Moreover, the number of rim curves are limited by the number of images/cameras. As a result, the edge of the contour polygons at places where the rims are very sparse will be very long. During reconstruction, long edges will lead to ill-formed triangles. In this paper, long edges are subdivided by inserting additional points along it. A more aggressive scheme is also possible: since we know the surface normal for each vertex of the contour polygon, long edges can be replaced with fitted parabola curves to make it look smoother. Since the surface normal at each vertex is known, we can make sure the contour polygons, and hence the final surface, always fall within the visual hull defined by the silhouettes, while making them look smoother and aesthetically pleasing. Extracting Surface Representations From Rim Curves.

3.2 Mesh Extraction from Cross-Section Contour

The major difficulty in the extraction of a mesh from cross-section contours is the branching problem [5]. Here, one reasonable assumption is made that the object is not extremely skew and the intercrossing planes are dense enough to present the topology.

Let us consider two adjacent contours, and denotes $\mathcal{A}_i$ the contour polygon on one layer and $\mathcal{W}_j$ the contour polygon on the layer immediately below.

Definition 1. *If and only if the center of $\mathcal{A}_i$ can be projected within $\mathcal{W}_j$ or the center of $\mathcal{W}_j$ can be projected within $\mathcal{A}_i$, $\mathcal{A}_i$ and $\mathcal{W}_j$ have a* ***connectedness relationship****.*

Definition 2. $\langle \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}, \{\mathcal{W}_1, \ldots, \mathcal{W}_n\} \rangle$ *is an* ***m-n connectedness pair*** *if and only if:*

- *Neither* m *nor* n *is 0,*
- *For any $\mathcal{A}_i \in \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$, there exists $\mathcal{W}_j \in \{\mathcal{W}_1, \ldots, \mathcal{W}_n\}$ that $\mathcal{A}_i$ and $\mathcal{W}_j$ are having a connectedness; For any $\mathcal{W}_k$ having a connectedness with any $\mathcal{A}_i \in \{\mathcal{A}_0, \ldots, \mathcal{A}_m\}$, $\mathcal{W}_k \in \{\mathcal{W}_0, \ldots, \mathcal{W}_n\}$ is true and vice versa.*

According to the definition above, the *1-1 connectedness pair* corresponds to **a simple structure**, while *others* are **branching structure** [10, 11] (see Fig. 1(a) and Fig. 2(a)).

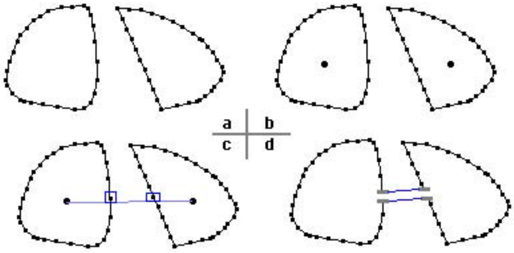


Fig. 1. (a) Contour data with branching structure. (b) Centers of the contour polygons. (c) Vertices to be slipped is selected by considering the line joining the two centers. (d) Conversion of the structure into a simple structure with the additional directed edge pair.

Branching structure is a challenging problem in contour reconstruction. The major difficulty is how to slip from one branch to another automatically. Further, the output mesh should be obturated.

To handle the branching structure, an additional directed edge pair is introduced. The additional directed edge pair is formed by two edges with the same ends but opposite directions. They are used to decide where to slip from one polygon into another. The aim is to emerge all the polygons on each side of the m-n connectedness pair. Here, we introduce a method to gain the additional directed edge pair.

First, consider a 2-1 branching structure (see Fig. 1). The centers of the two polygons that to be connected are first computed (see Fig. 1(b)). Two vertices each from one polygon are then selected. The vertex should be the one closest to the line segment connecting the centers of the two polygons (see Fig. 1(c)). Finally, a directed edge pair with these selected vertices as end points is added between the two polygons, and the branching structure is transformed into a simple structure (ss Fig. 1(d)).

Actually, the real world topology might be much more complex than a 2-1 branching structure. We solve it by recursively applying the above method. To do this, each time we pick two polygons on one slicing plane having the shortest center distance. These polygons form a 2-1 branching structure and can be solved by earlier mentioned method. We repeat the process of picking and solving a 2-1 branching structure in one slice until the *m-n branching structure* becomes a *1-n branching structure*. The same process is then applied to the other slice until the structure becomes a *1-1 structure* (simple structure).

Some polygons are formed from several polygons via additional directed edge pairs. If a mesh is extracted by a naive greedy algorithm that produces edges with minimum length at every step, the output mesh will be in an ill form (see Fig. 2(b)). To make the mesh obturated, the mesh has to be extract by the vertex sequence. On the other hand, to guarantee the maximal quality, all possible edges linking the vertices on the two cross-section contours should be considered. Thus, our optimization problem of identifying an energy minizing connectedness pairs perfectly fits into the context of dynamic programming techniques.

There are three kinds of mesh elements in the mesh belt between two polygons on adjacent contours, namely right triangle, invert triangle and quadrangle.

Definition 3. *Begin with two vertices having the shortest distance between two polygons, denotes:*

$pc_{i,j}$ *: the error of a* $right\ triangle_{<i,j>}$ *formed by the ith vertex on the polygon above and the edge* (j,j+1) *on the polygon below, defined as*

$$pc_{i,j} = |p_i, c_j|^2 ; \tag{1}$$

$cp_{i,j}$ *: the error of an* $invert\ triangle_{<i,j>}$ *formed by the edge* (i,i+1) *on the polygon above and the jth vertex on the polygon below, defined as*

$$cp_{i,j} = |c_i, p_j|^2 ; \tag{2}$$

$cc_{i,j}$ *: the error of a* $quadrangle_{<i,j>}$ *formed by the edge* (i,i+1) *on the polygon above and the edge* (j,j+1) *on the polygon below, defined as*

$$cc_{i,j} = 2 * |c_i, c_j|^2 ; \tag{3}$$

$\mathcal{E}_{i,j}$ *: the minimum error of the mesh belt from begin to the ith vertex of the polygon above and the jth vertex of the polygon below, defined as*

$$\mathcal{E}_{i,j} = \begin{cases} 0, & \textit{i=0, j=0;} \\ \mathcal{E}_{0,j-1} + pc_{0,j-1}, & \textit{i=0, j}\neq\textit{0;} \\ \mathcal{E}_{i-1,0} + cp_{i-1,0}, & \textit{i}\neq\textit{0, j=0;} \\ min\{\mathcal{E}_{i,j-1} + pc_{i,j-1}, & \\ \quad \mathcal{E}_{i-1,j} + cp_{i-1,j}, & \textit{otherwise} \\ \mathcal{E}_{i-1,j-1} + cc_{i-1,j-1}\} & \end{cases} \tag{4}$$

Note that for the case $i \neq 0$ and $j \neq 0$ in (4), if we set

$$\mathcal{E}_{i,j} = min\left\{\mathcal{E}_{i,j-1} + pc_{i,j-1}, \mathcal{E}_{i-1,j} + cp_{i-1,j}\right\} , \tag{5}$$

the outcome will be triangle-only scheme mesh with maximal quality as showed in Fig. 4(f).

To extract a mesh belt, we first computer all the $\mathcal{E}_{i,j}$, and record the corresponding values from $pc_{i,j}$, $cp_{i,j}$ and $cc_{i,j}$. By backtracking from the end to the beginning, the mesh belt with minimal error (maximal quality) could be extracted (see Fig. 2(c)). Next we scan the whole mesh belt and find out all pairs of mesh elements which involve the additional directed edge pair. If any of these mesh elements is a quadrangle, it will be divided along its shorter diagonal and converted into a triangle. The pair of mesh elements can thus always be converted to the form of two triangles with the additional directed edge being the common edge (see Fig. 2(c)). Finally, the directed edge is replaced by an edge joining the two opposite vertices of the two triangles (see Fig. 2(d)).

After the above process, there may still be some polygons that do not belong to any connectedness pair. Actually, they lie on the top/bottom layers of the real world model or the ends of branches. We simply close these polygons to make the final mesh water tight. To do this, concave polygons are first divided into convex ones. By dividing by its shortest diagonal, each convex polygon will become two smaller convex polygons recursively until all are triangles.

The complete process of extracting mesh from 3D contour structure is summarized in algorithm 1.

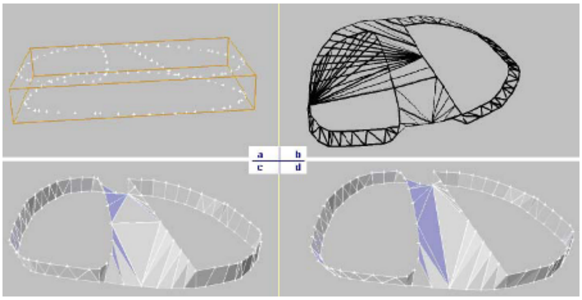


Fig. 2. (a) Contour data with branching structure. (b) Mesh extracted in local optimal by naive greedy algorithm. (c) Mesh extracted in global optimal using the proposed dynamic programming based algorithm. (d) Mesh after removing additional directed edge pair.

Algorithm 1. Mesh Extraction from 3D Rims

```
construct topologically correct cross-section contours;
construct the connectedness pair;
for all connectedness pairs do
    if the structure contains branching structure then
        convert the branching structure into simple structure;
    end if
    compute the minimal error with (4);
    record the choice of computation in each step;
    while backtracking the steps do
        recover the mesh element;
        move backwards;
    end while
    if the structure contains branching structure then
        reconstruct all the elements involving the additional directed edge pair;
    end if
end for
for all polygons not belong to any connectedness pair do
    repeat
        divide the polygon with the shortest diagonal;
    until all are triangles
end for
```

4 Experiments and Results

The first experimental sequence consists of rims recovered from 20 images from a turntable sequence of “Girl and Teddy” toy with fairly complex topology (see Fig. 3). The cameras are calibrated using a method proposed in [12]. The recovered 3D rims are sliced by 121 planes and this results in 6,207 vertices. After reconstruction, the maximal

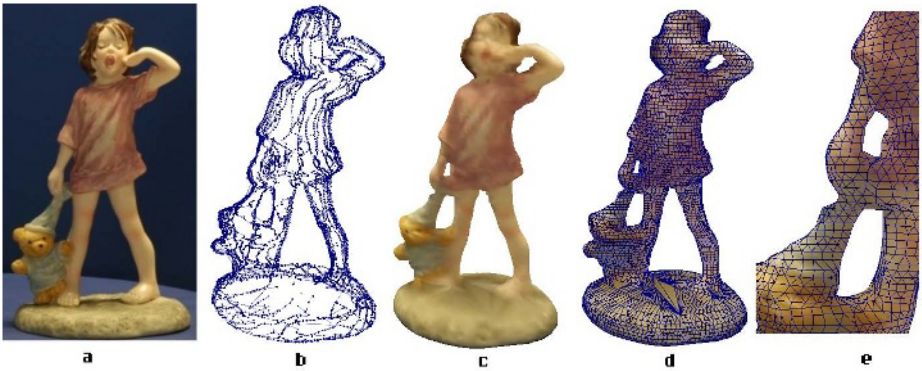


Fig. 3. Reconstruction of a girl and teddy toy with complex topology from a turntable sequence (20 images). (a) The original image. (b) Recovered 3D rims. (c) Resulting surface with texture mapping. (d) Resulting surface with the wire-framed mesh superimposed. (e) Local view of the mesh.

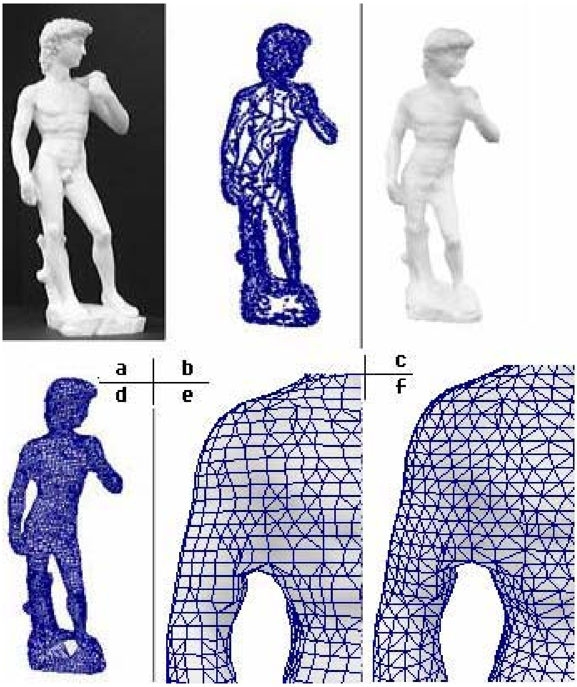


Fig. 4. Reconstruction of a David statuette from a turntable sequence (20 images). (a) The original image. (b) Recovered 3D rims. (c) Resulting surface with texture mapping. (d) Resulting surface with the wire-frame mesh superimposed. (e) Local view of mesh under triangle/quad mixed scheme. (f) Local view of mesh under triangle-only scheme.

quality mesh is formed by 5,805 triangles and 3,334 quadrangles. Figure 3(c) shows the final result after texture mapping from the original image.

The second experimental sequence consists of 3D rims recovered from 18 images of a turntable sequence of a David statuette (see Fig. 4). The recovered 3D rims are sliced by 110 contour planes and this results in 5,990 vertices. Figure 4(c) shows the final result after texture mapping from the original image. Figure 4(e) shows the optimal mesh under mixed scheme with 5,990 triangles and 3,116 quadrangles, comparing with the optimal mesh under triangle-only scheme with 11,980 triangles (see Fig. 4(f)).

5 Conclusions and Future Work

In this paper, we present a novel method for extracting a surface from 3D rims recovered from silhouettes. This method exploits the connectivity information implied by the rim curves to produce a set of topologically correct cross-sections, from which the final surface is extracted. The final surface is a mixed triangle/quadrangle scheme optimal mesh, which produces more regular yet feature-preserving meshes than using the traditional triangle-only mesh.

One limitation is that the geometric position of each vertex is fixed a priori to mesh extraction. Thus, the four vertices of a quadrangle obtained in the reconstruction may not be co-planar. We are now exploring a new algorithm to handle this problem which subdivides the extracted mesh according to the curvature and other local properties of the surface.

Acknowledgements

This project is partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project HKU 7155/03E.

References

1. Liang, C., Wong, K.-Y. K.: Complex 3d shape recovery using a dual-space approach. In: Proc. IEEE International Conference on Computer Vision and Pattern Recognition. Volume 2. (2005) 878–884
2. Boyer, E., Berger, M.: 3d surface reconstruction using occluding contours. International Journal on Computer Vision **22** (1997) 219–233
3. Vaillant, R., Faugeras, O.: Using extremal boundaries for 3-d object modeling. IEEE Trans. Pattern Analysis and Machine Intelligence **14** (1992) 157–173
4. Cong, G., Parvin, B.: Surface recovery from planar sectional contours. In: Proc. International Conference on Pattern Recognition. Volume IV. (2000) 106–109
5. Boissonnat, J.: Shape reconstruction from planar cross sections. Computer Vision Graphics and Image Processing **44** (1988) 1–29
6. O'Neill, B.: Elementary Differential Geometry. 2rd edn. Academic Press (1966)
7. Devillers, O.: On deletion in delaunay triangulations. In: Proc. 15th Annual Symposium on Computational Geometry. (1999) 181–188
8. Schaefer, S., Warren, J.: On c2 triangle/quad subdivision. ACM Trans. Graph. **24** (2005) 28–36

9. Stam, J., Loop, C.: Quad/triangle subdivision. Computer Graphics Forum **22** (2003) 79–85
10. Bresler, Y., Fessler, J., Macovski, A.: A bayesian approach to reconstruction from incomplete projections of a multiple object 3d domain. IEEE Trans. Pattern Analysis and Machine Intelligence **11** (1989) 840–858
11. Meyers, D., Skinner, S., Sloan, K.: Surfaces from contours. ACM Trans. Graph. **11** (1992) 228–258
12. Wong, K.-Y. K., Cipolla, R.: Structure and motion from silhouettes. In: Proc. 8th IEEE International Conference on Computer Vision. Volume II. (2001) 217–222

Applying Non-stationary Noise Estimation to Achieve Contrast Invariant Edge Detection

Paul Wyatt and Hiroaki Nakai

Multimedia Laboratory, Toshiba Corporate RDC, 1 Komukai-Toshiba-cho,
Saiwai-ku, Kawasaki 212-8582, Japan
wyatt@eel.rdc.toshiba.co.jp, hiroaki.nakai@toshiba.co.jp

Abstract. To recognize or identify objects it is desirable to use features which are minimally affected by changes in lighting and non-stationary noise. This requires accurate estimation of both signal and noise.

In response to this challenge, this paper proposes a method for estimation of non-stationary isotropic noise based on steering filters to directions perpendicular and parallel to the local signal. From the filter responses in this direction equations for signal and noise are obtained which lead to an edge detection method dependent solely upon local signal-to-noise ratio. The proposed method is compared to various common edge detection methods from the literature, on synthetic and real images. Quantitative improvement is demonstrated on synthetic images and qualitative improvement on real images.

1 Introduction

The extraction of edges and curves is of considerable interest to the vision community, as is evident from the large, diverse literature on the subject, e.g. [1, 2, 3, 4, 5, 6]. From the literature, perhaps three key ideas have emerged. Firstly, orienting filters according to some notion of local optimality: often defined as the direction in which the least squares energy is maximized [1]. Secondly, the idea that on an edge the responses to filters at different scales must be maximally in phase [1, 3] : essentially meaning that as an edge is an odd function, at an edge responses to odd filters will be maximal and even will be zero. The third idea is that structures should be associated with filter scale [4, 6, 7].

However, two related problems remain difficult: estimation of contrast change and non-stationary noise. Unaccounted for, both can lead to poor repeatability and instability. Contrast correction has been most successfully attempted using the Retinex transform [8]. However, it can err in dark regions. Possibly more promising is estimation of noise. If local signal s and noise σ_n are estimated, signal-to-noise ratio (SNR) can be established. This leads toward an edge measure such as $\left(1 - \frac{\sigma_n}{s}\right)$. Assuming both signal and noise share the same relation to contrast or illumination, this implies detected structures would be contrast invariant. The assumption is not unreasonable as edges occur where phase is congruent [1, 3] leading us to infer that signal power is significantly greater than noise power for the case of uncorrelated, isotropic noise. Consequently, even in

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 742–751, 2006.

poor contrast areas an edge requires that the ratio between signal and noise is significant.

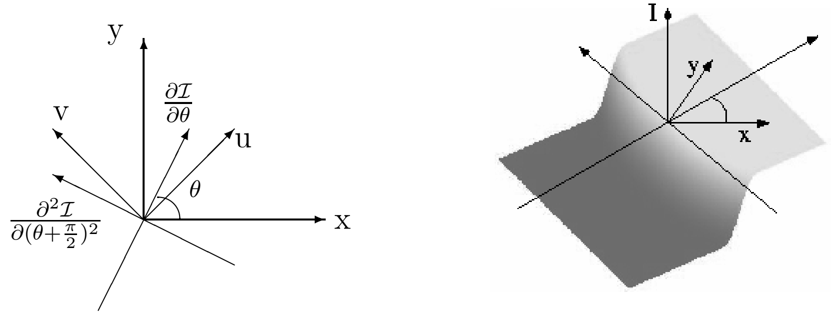
Approaches to noise estimation are often statistically based [3, 9, 10]. They assume that a high pass filtered image contains solely noise coefficients [3] and from an assumption on the expected noise distribution estimate noise variance [3, 11]. Other approaches have attempted to account for structure, for example through anisotropic evolution of the intensity [6]. However, this approach still requires an initial noise estimate. More recent alternatives have attempted to suppress structure, estimating noise from the remainder [12].

This paper focuses on estimation of non-stationary, uncorrelated isotropic noise as part of the structure detection process. Although the focus is on edge detection, the method is general and could equally well be applied to other types of feature, e.g. corners. We make two contributions. Firstly, an integrated model for simultaneously estimating the local noise and structure to obtain an edge measure dependent solely upon the SNR. This improves stability to contrast change. Secondly, we show how scale affects the problem of noise estimation and its applicability to distinguishing a step edge from shadowing.

2 A Combined Edge and Noise Model

This section first focuses on the edge and noise model, considering a single scale. It then proceeds to consider the differences for edges at different scales; i.e. with different degrees of blurring.

Figure 1 contains two diagrams showing an abstraction of a generic edge, a step change in intensity $\mathcal{I}$, positioned at angle θ to image axes x and y. Axes u and v are aligned at $\frac{\pi}{4}$ to x and y. $\frac{\partial \mathcal{I}}{\partial \theta}$ and $\frac{\partial^2 \mathcal{I}}{\partial (\theta + \frac{\pi}{2})^2}$ denote two of the partial derivatives of $\mathcal{I}(x, y)$ perpendicular (orthogonal) and parallel to the edge, our third co-ordinate system. It is assumed that non-stationary Gaussian noise $\mathcal{N}(0, \sigma_n^2)$ is present. It has been shown, see [1, 3], that a one dimensional edge is comprised


y
v
$\frac{\partial \mathcal{I}}{\partial \theta}$
u
θ
x
$\frac{\partial^2 \mathcal{I}}{\partial (\theta + \frac{\pi}{2})^2}$
I
y
x

(a) Plan schematic (b) Isometric view

Fig. 1. Schematic for a general curve-like point. The derivatives are derivatives of the intensity function perpendicular and parallel to the local edge structure.

solely of odd sinusoids: a sum which is theoretically in phase at the location of the step. This sum must be large relative to σ_n to facilitate detection. Conversely, the sum of co-sinusoids will be small. These properties can be stated mathematically:

$$\left|\frac{\partial \mathcal{I}}{\partial \theta}\right| \gg \sigma_n \ , \left|\frac{\partial^2 \mathcal{I}}{\partial \theta^2}\right| \sim \left|\mathcal{N}(0, \sigma_n^2)\right| \ . \tag{1}$$

Moving perpendicularly away from the edge (axis θ), equations 1 do not hold. The second derivative $\frac{\partial^2 \mathcal{I}}{\partial \theta^2}$ will increase, making this pair unsuitable for noise estimation. However, equations 1 consider behaviour solely in spatial direction θ. Derivatives in directions θ and $\theta + \frac{\pi}{2}$ are related via curvature κ:

$$\kappa = \frac{\frac{\partial \mathcal{I}}{\partial \theta}\frac{\partial^2 \mathcal{I}}{\partial(\theta+\frac{\pi}{2})^2} - \frac{\partial \mathcal{I}}{\partial(\theta+\frac{\pi}{2})}\frac{\partial^2 \mathcal{I}}{\partial \theta^2}}{\left[\left(\frac{\partial \mathcal{I}}{\partial \theta}\right)^2 + \left(\frac{\partial \mathcal{I}}{\partial(\theta+\frac{\pi}{2})}\right)^2\right]^{\frac{3}{2}}} \approx \frac{\frac{\partial \mathcal{I}}{\partial \theta}\frac{\partial^2 \mathcal{I}}{\partial(\theta+\frac{\pi}{2})^2}}{\left[\left(\frac{\partial \mathcal{I}}{\partial \theta}\right)^2 + \left(\frac{\partial \mathcal{I}}{\partial(\theta+\frac{\pi}{2})}\right)^2\right]^{\frac{3}{2}}} \tag{2}$$

as $\frac{\partial^2 \mathcal{I}}{\partial \theta^2} \sim 0$. As $\frac{\partial^2 \mathcal{I}}{\partial(\theta+\frac{\pi}{2})^2}$ varies, the local structure changes from a straight line to more tightly curved structures (eventually corner-like). Assuming curvature varies smoothly then $\frac{\partial \mathcal{I}}{\partial(\theta+\frac{\pi}{2})}$ should be relatively small. For the case where $\kappa = 0$, a straight edge, $\frac{\partial^2 \mathcal{I}}{\partial(\theta+\frac{\pi}{2})^2} = 0$. In this case, if $\frac{\partial \mathcal{I}}{\partial(\theta+\frac{\pi}{2})} \neq 0$, it must be responding to some disturbance. *We make the assumption that the measurement* $\frac{\partial \mathcal{I}}{\partial(\theta+\frac{\pi}{2})}$ *must be related to the local noise* σ_n. Generally, for any smoothly curving structure, it can reasonably be assumed that this will be true. With equation 2 this observation completes our edge model:

$$\frac{\partial^2 \mathcal{I}}{\partial(\theta + \frac{\pi}{2})^2} \propto \kappa \text{ , and} \tag{3}$$

$$\frac{\partial \mathcal{I}}{\partial(\theta + \frac{\pi}{2})} \sim \mathcal{N}(0, \sigma_n^2) \ . \tag{4}$$

Although equation 1 is directly suitable for implementation, equation 4 is not. It yields, not σ_n, but one sample from a noise distribution at each point. An estimate of σ_n, $\hat{\sigma_n}$ can be made using samples from a small region about each point. Using δ to denote the extent of this area, about point $x = x_i, y = y_i$,

$$\hat{\sigma}_n \approx \sqrt{\frac{1}{4\delta^2}\int_{x_i-\delta}^{x_i+\delta}\int_{y_i-\delta}^{y_i+\delta}\left(\frac{\partial \mathcal{I}}{\partial(\theta + \frac{\pi}{2})}\right)^2 dydx} \ . \tag{5}$$

Note that $E[\frac{\partial \mathcal{I}}{\partial(\theta+\frac{\pi}{2})}]$ is expected to be zero and is therefore not required for the estimation of $\hat{\sigma}_n$ in equation 5. δ should be set in proportion to the spatial extent of the derivatives filter. Practically a value of twice the largest filter dimension in pixels is used. Furthermore, note that $\hat{\sigma}_n^2$ is scaled by the filter used in obtaining $\frac{\partial \mathcal{I}}{\partial(\theta+\frac{\pi}{2})}$. $\hat{\sigma}^2 \sim \sigma_{true}^2 \sum_i f_i^2$ where f_i are the filter coefficients.

This becomes important if more than one filter scale is used. Using the estimate of signal energy $\frac{\partial \mathcal{I}}{\partial \theta}$ and $\hat{\sigma_n}$, the edge detection measure $\Pr(S)$ is then simply

$$\Pr(S) = \left(\frac{\left\lfloor \left| \frac{\partial \mathcal{I}}{\partial \theta} \right| - \alpha \sigma_n \right\rfloor}{\left| \frac{\partial \mathcal{I}}{\partial \theta} \right|} \right) , \tag{6}$$

where $\lfloor f() \rfloor$ bounds the function, $f()$, from below with zero. As the energy response function is statistical in nature, it suppresses a fraction of all noise responses. For example, setting $\alpha = 1.6449$ will suppress 90% of noise. Practically, α can be adjusted according to whether it is preferable to suppress noise or obtain every potential structure. Results in this paper use $\alpha = 2.5$.

2.1 Natural Scale and Edge Detection

As the model we have defined assumes that fixing filters at one scale is sufficient for edge detection, we briefly justify this with respect to scale invariance. Scale invariance, see [4], shows that a filter has a size or scale by which it may be parameterized and normalisation by this parameter makes the response independent of scale. For instance, the one dimensional family of Gaussians $\frac{\partial^n \mathcal{G}(x)}{\partial x^n}$, $n = 0, 1, 2$, can be parameterized by standard deviation γ. Scale invariance is achieved through multiplication by γ^n.

For an ideal edge, with no blur, the response of $\gamma \partial \mathcal{G}$ will theoretically be constant until γ increases such as to cause interaction with a second edge. After this the response decreases [4]. In practice, for $\gamma < 2$, the filter approximation tends to yield a quickly rising step which plateaus between $1.5 \leq \gamma \leq 3$. Consequently, for an ideal edge using a filter with $\gamma \geq 2$ should yield constant results. For shadowed edges, with pre-blur γ_s, the response of $\gamma \partial \mathcal{G}$, $\gamma \leq \gamma_s$, grows linearly plateauing at $\gamma = \gamma_s$. Consequently, we can detect (and remove) shadowed edges by comparing the ratio of coefficients at two or more different scales. This part of the contrast problem is not further examined in this paper.

3 Implementation

Having detailed a model for an edge in non-stationary noise along with the requisite equations for estimating these properties we now detail the approximations made between the theoretical model and its practical implementation. For context, we first state the complete algorithm.

1. Convolve the image with a Gaussian, of deviation $\gamma_1 = 2$. Calculate $\partial \mathcal{I}$ in directions x, u, y and v.
2. Estimate the global noise, σ_g, using an Expectation Maximization (EM) on the magnitudes of $\partial \mathcal{I}$ in directions x and y to obtain weights (ω_1, ω_2) and variances (σ_1^2, σ_2^2) for a Gaussian Mixture Model (GMM) of signal and noise. $(\mu_1 = \mu_2 = 0.)$
3. At each point: estimate θ and use it to select signal and noise samples from amongst the four derivatives.

4. At each point: evaluate equations 5 and 6.
5. If a binary edge map is desired, threshold using an estimate of the existing fraction of edges, $\min(\omega_1, \omega_2)$, from the EM algorithm.

Considering the filters, normal image blur normally has deviation less than 2 pixels: the Gaussian smoothing filter's is set to the same. It is truncated at 2 deviations. Derivatives are simple central differences, with the diagonal directions scaled appropriately by $\frac{1}{\sqrt{2}}$. The separation of filter responses into noise and signal is then achieved using the least squares estimate of θ: $\theta = \arctan\left(\frac{\partial \mathcal{I}}{\partial y} / \frac{\partial \mathcal{I}}{\partial x}\right)$ [13]. For $\frac{\pi}{8} \leq \theta \leq \frac{3\pi}{8}$ and for $\frac{5\pi}{8} \leq \theta \leq \frac{7\pi}{8}$ derivatives along axes u and v are used, otherwise along x and y. With respect to the choice of filters, first derivatives in four directions is simple and suffices. Although equations 1,3 and 4 are specified in terms of the edge co-ordinate system θ and $\theta + \frac{\pi}{2}$, the question of how to estimate this is not simple. Although other methods for steering filters exist, e.g. [2], they can be computationally expensive in practice and the complex steering mechanism can induce errors for small (7x7 pixels) filters. However, using only four filters, responses away from the axes are affected by image quantisation. For example, edges oriented at $\theta = \frac{(2n+1)\pi}{8}, n = 0, 1, 2$ fall directly between the filter directions. Noise in these directions will be over-estimated leading to decreased stability. If speed is less important than accuracy, more complex steering techniques could be used [13, 2].

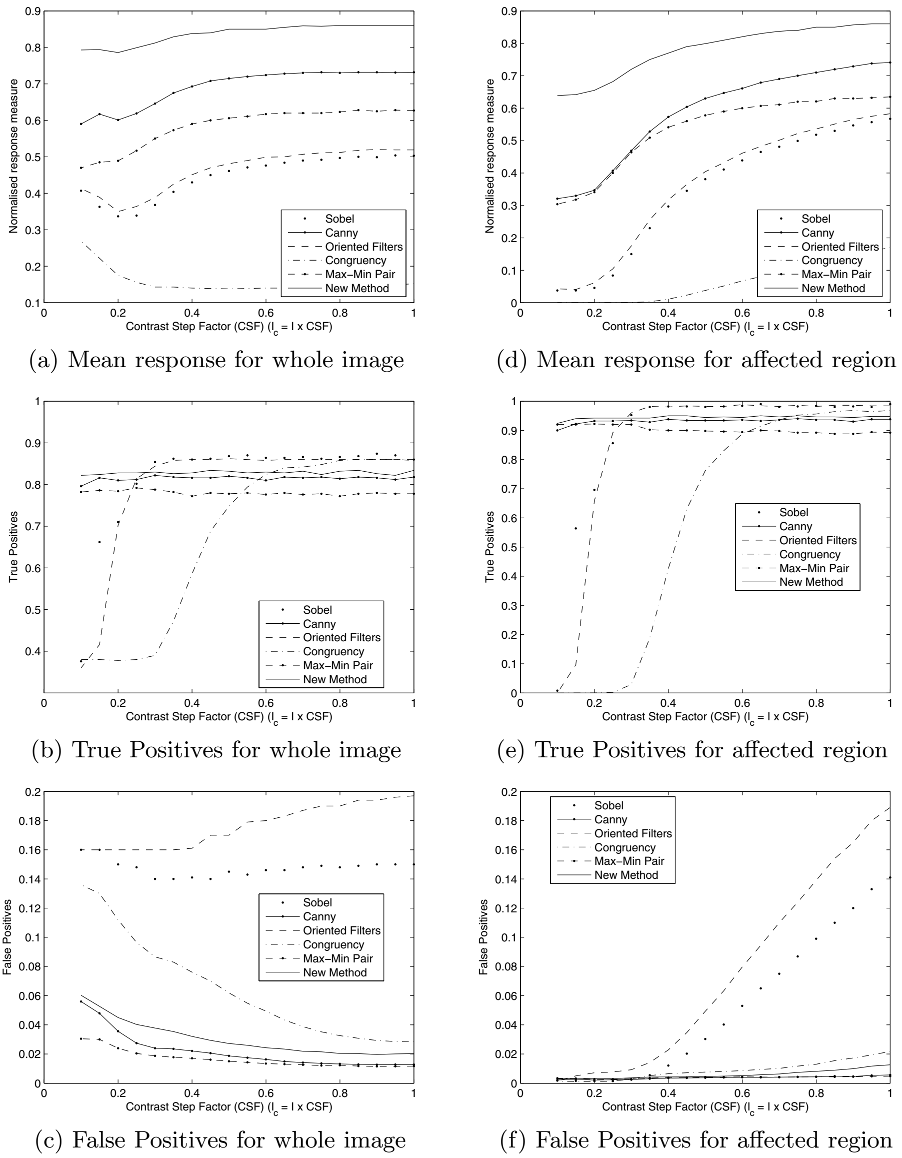
Equation 6 is calculated as stated. Two practical changes are made to 5. Firstly, the integral over the image grid is replaced by convolution with a Gaussian. This second Gaussian's deviation γ_2 is set at twice that of the Gaussian used for the first image convolution: $\gamma_2 = 2\gamma_1 = 4$. Secondly, if $\hat{\sigma_n} \leq \frac{\sigma_g}{20}$ $\hat{\sigma}_n$ is set equal to the global noise.

Finally, estimation of σ_g models wavelet coefficients as being comprised, at each scale, by a two state GMM [11]. One state has large variance and denotes structure, the other small variance and denotes noise. Both have zero mean. This idea is also used in wavelet based denoising [10]. The parameters for this model are fitted using an EM algorithm, yielding σ_g and weights, ω_1, ω_2, for the fractional split between edges and noise. The threshold for equation 6 is chosen to obtain this fraction of image points as edges. The EM algorithm is selected simply as it is one method appropriate for fitting a model to unlabeled data.

4 Experimental Results

Our method has been evaluated on a wide range of images; real and synthetic. As ground truth is difficult to establish for real images, the performance of the noise estimation is established on synthetic images. Sample results from many tested real images are given.

In our first test, synthetic images containing a mixture of curved and straight lines were created. To these, noise with a deviation equal to $\frac{1}{4}$ the edge size is added. Note that the image's left and right hand halves are exact duplicates. Then, a contrast step is applied to one half of the image. This yields images



(a) Mean response for whole image

(d) Mean response for affected region

(b) True Positives for whole image

(e) True Positives for affected region

(c) False Positives for whole image

(f) False Positives for affected region

Fig. 2. Evaluation of common measures on Synthetic Test Images containing various oriented and curved structures

with varying contrast but constant SNR: suitable to test whether a method is stable with respect to contrast change. As the images are synthetic, ground truth is known. Various measures are evaluated; True positives: number pixels correctly classified; False positives: number incorrectly classified and, the edge detection energy (equation 6) at an edge. The tests were repeated to remove any statistical bias from a particular set of generated noise. Stability was evaluated for Sobel, Canny, oriented bandpass filters, phase congruency, pseudo-steering through taking the maximum and minimum of four oriented filters to be signal and noise respectively (Max-Min) and the proposed edge detection methods. Results are shown in figure 2. As can be seen from figure 2, (a) and (d), the proposed method produces a more stable edge energy measure with respect to contrast change than the tested alternatives. Although it experiences some disturbance, for contrast step factors (CSF) ≤ 0.4, this is smaller than for the alternatives. True positives for most methods remain stable down to a CSF of 0.1. In this respect, no method is clearly better. A final point is that increases in false positives in 2(c) reflects the fact that global noise will be underestimated in the presence of significant contrast changes across the image. The proposed method is also affected, despite estimating noise locally, as it will make a percentage of errors in regions where there are no *real* edges: steering the estimation to the direction perpendicular to the strongest response (which in these regions is noise) leads to under-estimation of noise.

Our second set of tests repeats the experiment on synthetic images with real images to which noise and contrast steps were applied. Two sample images from amongst these are shown in figure 3;from the 'Bad Etting' sequence, available at iw1www.ira.uka.de/image_sequences, and the 'Graffiti' data set, available at www.inrialpes.fr/lear/people/Mikolajczyk/). The only difference from the first test, is that after adding noise, the images were requantised to 8 bits. The edge energy functions for the Canny edge function and the proposed method are shown for these images in figures 4 and 5. The Canny edge function is used as the comparison simply as it is probably the most widely used and available

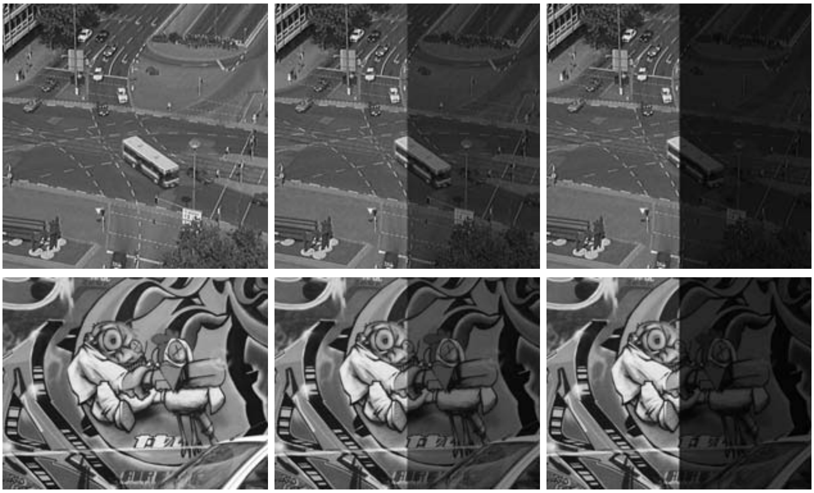
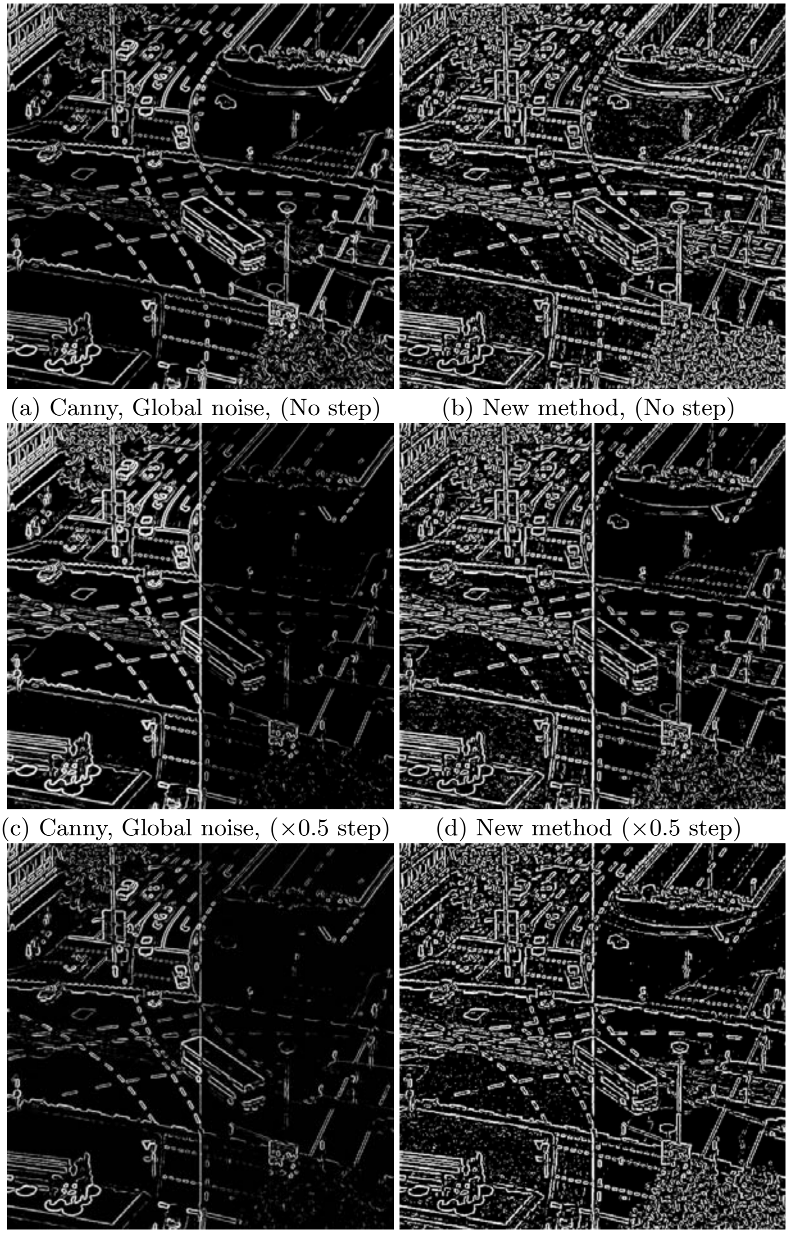


Fig. 3. The image pairs, with contrast steps, used for the tests shown in fig. 4 and 5



(a) Canny, Global noise, (No step)

(b) New method, (No step)

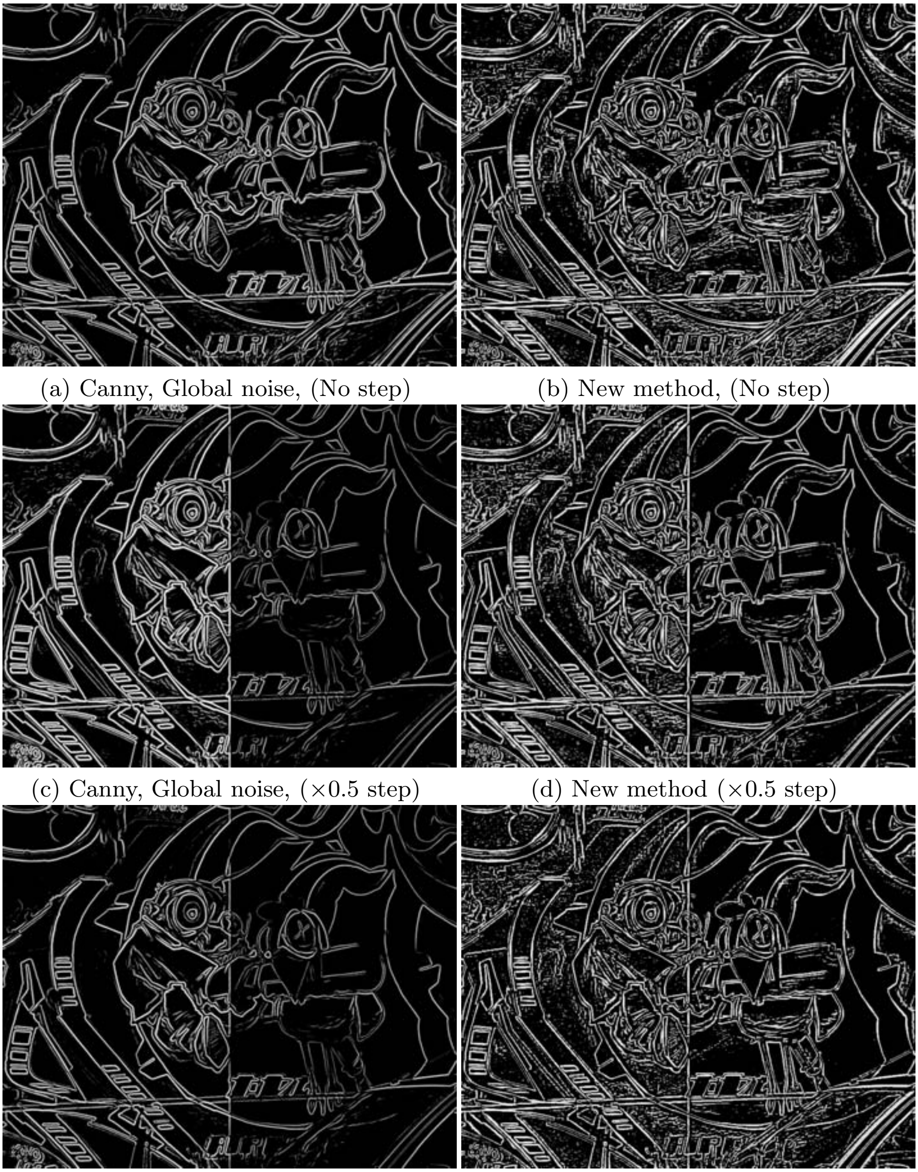
(c) Canny, Global noise, (×0.5 step)

(d) New method (×0.5 step)

(e) Canny, Global noise, (×0.25 step)

(f) New method (×0.25 step)

Fig. 4. Test set 1: Comparison of Edge detection methods. The two columns show the new method versus the standard Canny edge function, for the original image and with contrast steps of ×0.5 and ×0.25 applied. Images additionally had 2% Gaussian isotropic noise added, *prior* to the contrast step being applied. Note that the images show the edge response energies, and are *NOT* binary edge images.



(a) Canny, Global noise, (No step)

(b) New method, (No step)

(c) Canny, Global noise, (×0.5 step)

(d) New method (×0.5 step)

(e) Canny, Global noise, (×0.25 step)

(f) New method (×0.25 step)

Fig. 5. Test set 2: Edge detection methods for original image and with varied contrast and noise. The two columns show detection the new method versus the standard Canny edge function, for the original image and with contrast steps of ×0.5 and ×0.25 applied. These images have additionally had 2% Gaussian isotropic noise added, *prior* to the contrast step being applied. Note that the images show the edge response energies, and are *NOT* binary edge images.

method. As can be seen, the energies obtained from the proposed method are noticeably more stable with respect to contrast change. A final point is that weak structure can vanish with requantisation: for $CSF = \frac{1}{\delta I}$, a step of less than $\frac{1}{2}\delta I$ becomes constant. Noise magnifies this effect. Generally, from testing on various sequences, e.g. traffic, people, the method appears stable and reliable. Improvement relative to a standard method, e.g. Canny, depends upon whether contrast changes are present and whether the image quantisation has left the structure intact.

5 Summary and Conclusions

We have presented a method for incorporating local noise estimation into edge detection, thereby improving resilience to illumination change. Testing on synthetic and real data demonstrated improvement over previous methods.

References

1. Canny, J.: A computational approach to edge detection. IEEE Transactions PAMI **8** (1986) 679–698
2. Felsberg, M., Sommer, G.: The Monogenic Signal. IEEE Transactions Signal Processing **49(12)** (2001) 3136–3144
3. Kovesi, P.: Image Features from Phase Congruency. Videre: Journal of Computer Vision Research **1(3)** (1999) 1–27
4. Lindeberg, T.: Edge Detection and Ridge Detection with Automatic Scale Selection. IJCV **30(2)** (1998) 117–153
5. Pellegrino, F., Vanzella, W., Torre, V.: Edge Detection Revisited. IEEE: Systems, Man and Cybernetics **34(3)** (2004)
6. Perona, P., Malik, J.: Scale-space and Edge Detection Using Anisotropic Diffusion. IEEE Transactions PAMI **12(7)** (1990) 629–639
7. Elder, J., Zucker, S.: Local Scale Control for Edge Detection and Blur Estimation. IEEE Transactions PAMI **20(7)** (1998) 699–716
8. Adjeroh, D.: On Ratio Based Color-Indexing. IEEE Transactions Image Processing **10(1)** (2001) 36–48
9. Olsen, S.: Noise variance estimation in images. Graphic Models and Image Processing **55(4)** (1993) 319–323
10. Starck, J., Murtaugh, F.: Automatic Noise Estimation from the Multiresolution Support. Publ. of the Astronomical Soc. of the Pacific **110** (1998) 193–199
11. Crouse, M., Nowak, R., Baraniuk, R.: Wavelet-Based Statistical Signal Processing Using Hidden Markov Models. IEEE Transactions Signal Processing **46(4)** (1998) 886–902
12. Corner, B., Narayanan, R., Reichenbach, S.: Noise estimation in remote sensing imagery using data masking. J. Remote Sensing **24(4)** (2003) 689–702
13. Freeman, W., Adelson, E.: The design and use of steerable filters. IEEE Transactions PAMI **13(9)** (1991) 891–906

Corner Detection Using Morphological Skeleton: An Efficient and Nonparametric Approach

R. Dinesh and D.S. Guru

Department of Studies in Computer Science, University of Mysore,
Manasagangotri, Mysore-570 006, Karnataka, India
dinesh_r21@yahoo.com, guruds@lycos.com

Abstract. In this paper we propose an effective and robust approach for detecting corner points on a given binary image. Unlike other corner detection methods the proposed method is non-parametric in nature, that is, it does not require any input parameter. The proposed method is based on mathematical morphology. It makes use of morphological skeleton for detecting corner points. Convex corner points are obtained by intersecting the morphological boundary and the corresponding skeleton, where as the concave corner points are obtained by intersecting the boundary and the skeleton of the complement image. Experimental results show that the proposed method is more robust and efficient in detecting corner points.

Keywords: Corner points, Morphological skeleton, Non-parametric.

1 Introduction

Corner points on shape curves are effective primitives for shape representation and analysis. Corner points on a digital boundary are found at locations where the nature of the boundary changes abruptly and significantly. They provide critical information of a shape which is useful in pattern analysis and recognition problems. Therefore corner detection in images is an important aspect of computer vision applications.

Many algorithms have been developed for detecting corner points on the boundary curve of an object. The existing corner detection schemes can be broadly classified into two classes, (i) boundary based methods and (ii) Morphology based methods. The former methods involve computation of curvature at every point over a small segment of curve called region of support within the vicinity of the point of interest. The main problems of the boundary based methods are that they are computationally less efficient as they involve determination of region of support and then computation of curvature over the determined region of support involving lots of floating point operations. On the other hand the methods based on mathematical morphology are computationally efficient as they involve only integer operations. In view of this many methods based on mathematical morphology have been proposed in the literature.

[5] have proposed a method for corner detection using mathematical morphology, which makes use of morphological residue for detecting corner points. In their method convex and concave corner points are detected separately. Convex corner clusters are obtained by performing $(A - (A \circ B))$ operation. Subsequently boundary

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 752 – 760, 2006.

of the corner cluster is obtained by intersecting the obtained corner clusters with the original boundary. A convex corner point is obtained for each corner cluster. Corner point corresponding to a corner cluster is defined to be the point of intersection of the boundary of the cluster and the normal line passing through the center of the corresponding corner cluster. To detect concave corner points, the procedure is repeated on the complement image. Though the approach is fast, the boundary point detected as corner point is not necessarily a true corner point and some time they are shifted from the actual location. To overcome this problem [2] have suggested a modification by redefining the process of shrinking a corner cluster into a single point. In their approach a corner point in a cluster is defined to be the point with maximal N-hit number. Though the result of the method proposed by [2] is superior to that of [5], it fails to detect all corner points, as the method assumes the existence of only one corner point on each cluster, but in reality there may be more than one corner points in each corner cluster. In addition their method is sensitive to the boundary noise as the N-hit number is susceptible to noise.

[1] proposed a method for corner detection using the asymmetrical closing which is defined as a dilating of an image by a structuring element followed by eroding the result with another structuring element. The idea is to make the dilation and erosion complementary and correspond to variant types of corners. The disadvantage of this method is that sometimes it misses obtuse-angled corners and sharp-angled corners. [4] have proposed a corner detection algorithm based on the morphological skeleton. In this method the corner points are obtained by detecting the zero radius of the maximum plate on the morphological skeleton. The result of the corner detection is achieved by using logical hetero-OR operation between two corner sets of the source image and its complement set. But the drawback of this method is that it detects more spurious corner points. [3] have proposed an improvement to [1]. They have also proposed a method for corner detection using modified regulated morphological operators with adjustable strictness parameters. However, the method is sensitive to the value of strictness parameter, which is difficult to select.

In view of this, in this paper we propose a novel technique for detecting corner points which is as effective as any boundary based technique in detecting corner points and as efficient as any morphology based techniques. The proposed method detects convex and concave corner points separately. At first, morphological skeleton is obtained for a given image. Convex corner points are obtained by intersecting the obtained morphological skeleton of the image with its boundary. Similarly, concave corner points are obtained by performing the same set of operations on the complement image. Further, corner points of the complete image are obtained by performing the union of convex and concave corner points. Even though the method is based on mathematical morphology, the results of the proposed method is independent of size and shape of the structuring element and hence the method does not require any input parameter to decide the size and shape of the structuring element, thus the method is non-parametric in nature.

Rest of the paper is organized as follows: the proposed method is explained in section 2. Experimental results are presented in section 3 followed by the discussions in section 4. Finally paper concludes in section 5.

2 Proposed Method

In this section, we describe the proposed method for detecting corner points present on a given binary image. The proposed method has two stages, skeletonization of the given image followed by localization of corner points.

2.1 Skeletonization

Skeletonization is a process of reducing foreground regions in a binary image to a skeletal residue that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels. To realize this, imagine that the foreground regions in the input binary image are made of some uniform slow-burning material. Ignite simultaneously at all points along the boundary of this region and watch the fire move towards the centre of the image. At the points where the fire traveling from two different boundaries meet, the fire will extinguish itself. Such points put together form the so called `quench line' and this line is taken as the skeleton of the image. From the above definition it is clear that thinning of an image produces a sort of skeleton.

Skeletonization of an image is also defined as the loci of centers of bi-tangent circles that fit entirely within the foreground region of the image and Fig-1 illustrates this for a rectangular shape.

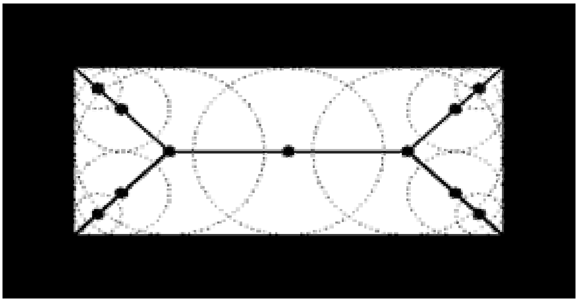


Fig. 1. Skeleton of a rectangle defined in terms of bi-tangent circles

Though, the terms medial axis transform (MAT) and skeletonization are often used interchangeably, in a strict sense, skeletonization is defined for a binary image and MAT is defined for a graylevel image, where each point on the skeleton has an intensity which represents its distance to a boundary in the original object.

The skeleton/MAT can be produced in two ways. The first is to use some kind of morphological thinning that successively erodes away pixels from the boundary (while preserving the end points of line segments) until no more thinning is possible, at which point what is left approximates the skeleton. The alternative method is to first calculate the distance transform of the image. The skeleton then lies along the *singularities* (*i.e.* creases or curvature discontinuities) in the distance transform. This latter approach is more suited for calculating the MAT since the MAT is same as the distance transform but with all points off the skeleton suppressed to zero.

The skeleton of a given image A can be expressed in terms of erosions and openings. That is, skeleton S of A denoted by S(A), is given by

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

with

$$S_k(A) = \bigcup_{k=0}^{K} \{(A \Theta kB) - [(A \Theta kB) \circ B]\}$$

where B is a structuring element, (A Θ kB) indicates k successive erosions of A; that is

$$(A \Theta kB) = ((\ldots (A \Theta B) \Theta B \Theta \ldots) \Theta B$$

k times, and K is the last iterative step before A erodes to an empty set. In other words,

$$K = \max\{k \mid (A \Theta kB) \neq \varnothing\}.$$

It can be noticed that A can be reconstructed from these subsets by using the equation

$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB).$$

2.2 Corner Localization

The proposed corner localization scheme detects convex and concave corner points separately. In order to obtain the convex corner points, the proposed method first extracts the morphological skeleton of the given image, and then the obtained morphological skeleton is intersected with the boundary of the image. Since morphological skeleton always touches the convex corner points, the intersection of morphological skeleton with the boundary helps us in detecting all convex corner points. Therefore, convex corner points are obtained as follows:

Let C be the boundary curve of the image A, which is obtained by

$$C = [A - (A \Theta B)]$$

where, B is 3 × 3 square structuring element.

Convex corner points are obtained by

$$\textit{Convex\_Corner (A)} = C \cap S(A)$$

Similarly, the concave corner points are obtained by

$$Concave\_Corner(A) = C \bigcap S(A^1 \oplus E)$$

where, A^1 is the complement image of A, and C is the boundary of A^1 and E is a rhombus structuring element. In case of detecting concave corner points, before obtaining the skeleton, A^1 is dilated with a rhombus structuring element E, because $S(A^1)$ does not have any point common with the C.

Overall corner points are obtained by performing union of convex corner set and concave corner set, that is,

$$Corner\ Points\ (A) = Convex\_Corner(A) \cup Concave\_Corner(A)$$

The proposed method is explained in detail with the help of the following illustration, Fig-2(a) shows the original image. Fig-2(b) shows the boundary of the original image and its morphological skeleton. It can be observed that morphological skeleton intersects the boundary of the image at convex corner points and hence the convex corner points are the points where the skeleton of the image intersects with the boundary of the image (Fig-2(c)). Fig-2(d) shows the boundary of the original image and the morphological skeleton for the complement of the original image. It can be observed that the skeleton of the complement image intersect the boundary at concave corner point. The detected concave corner points are given in Fig-2(e), and the corner points on the image, which is the union of convex and concave corner points is shown in Fig-2(f).

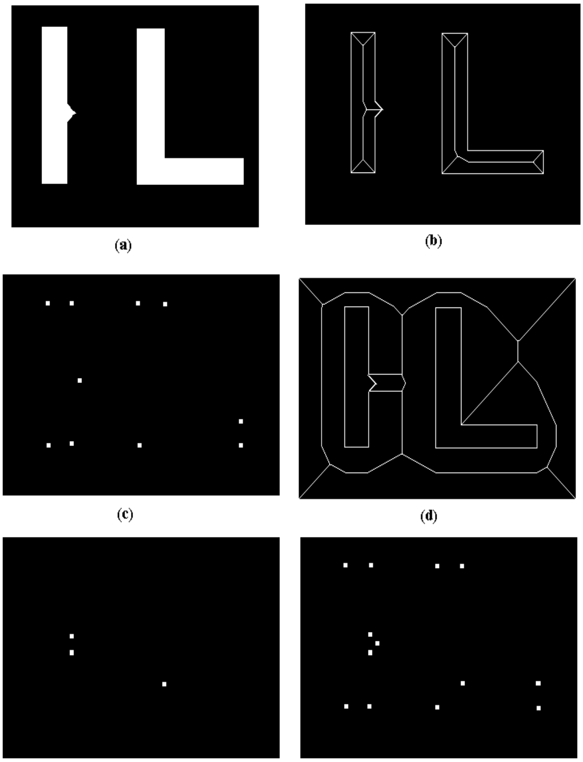


Fig. 2. (a) original image, (b) Boundary and skeleton of (a), (c) Convex Corner points, (d) skeleton of complement image, (e) Concave corner points and (f) All corner points

Thus, the proposed algorithm for detecting corner points on a given binary image is as trivial as follows:

```
Algorithm: Detect Corner Points
Input: Binary Image (A)
Output: Corner points of the image (A)
Method:
    Step 1: Extract the Boundary C as, $C = A - (A \Theta B)$.
    Step 2: Extract the Morphological Skeleton $S(A)$.
    Step 3: Obtain Convex corner points as Convex_Corners $(A) = C \cap S(A)$.
    Step 4: Extract the Morphological Skeleton $S(A^1 \oplus E)$.
    Step 5: Obtain Concave corner points as Concave_Corners $(A) = C \cap S(A^1 \oplus E)$.
    Step 6: Obtain Corner points (A) by performing union of Convex and Concave
            corner sets.
Algorithm ends.
```

It can be observed that the proposed method for detecting corner points does not require any input parameters, and hence it is non-parametric. Further, for detecting corner points it is not necessary to obtain the complete skeleton. Thus, it is quite enough if we apply four or five iteration of skeletinization process. In addition, efficiency of the proposed method further be improved by implementing the proposed method on parallel computers with SIMD architecture, where the process of convex corner and concave corner can be run parallely. Therefore the proposed method is both effective and efficient.

3 Experimental Results

In order to reveal the robustness of the proposed method in real pragmatic situation we have conducted several experiments on several shapes (including the shapes

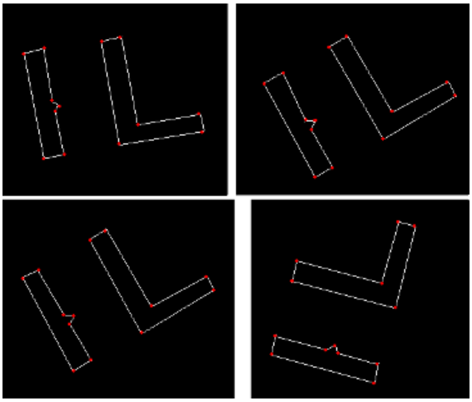


Fig. 3. Results of the proposed method for shape shown in Fig-2(a), in different orientations

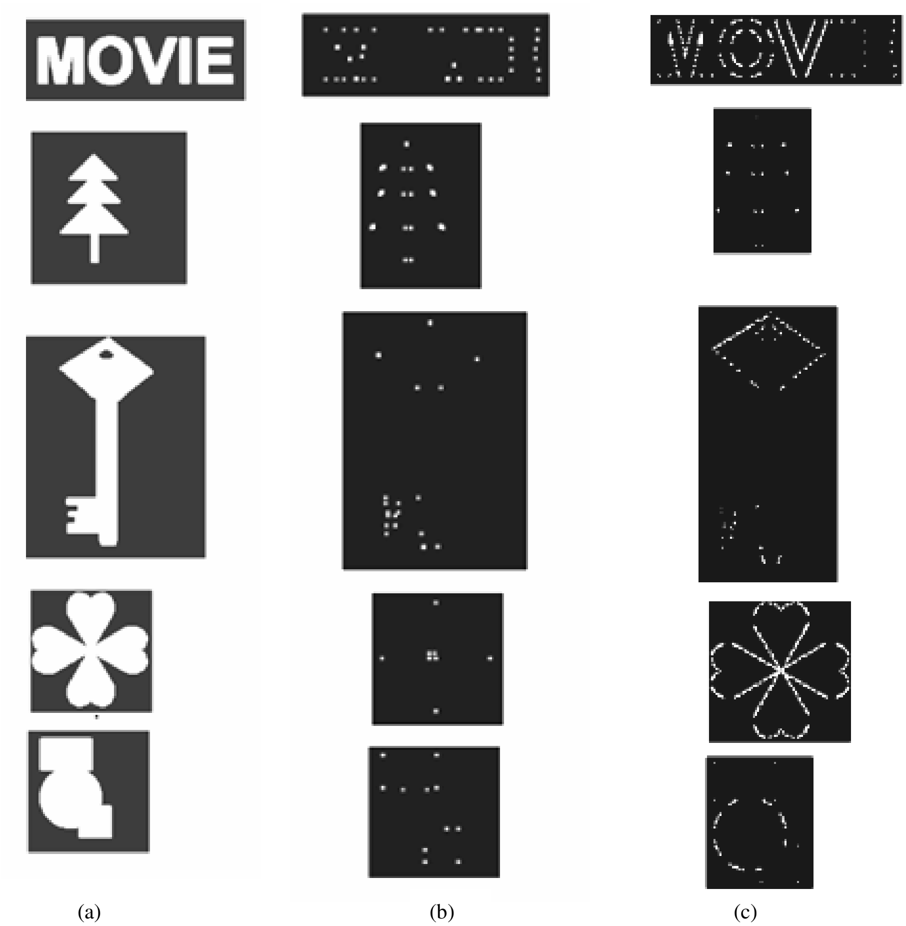


Fig. 4. (a) Input Objects, (b) Corner points detected by proposed method and (c) Corner points detected by Yu et. al., (2001)

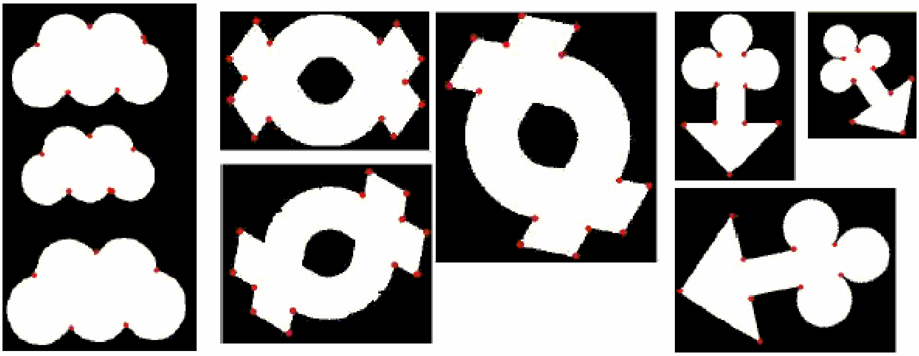


Fig. 5. Results of the proposed method for shapes in different orientations and scaling factors

considered by many researchers) with different scaling factors and in different orientations. The set of shapes considered for experimentation includes shapes with smooth curve segments of different radii or curvature, curvilinear segments, straight line segments. Fig-4 (b) shows the results of the proposed method for various shapes. For the purpose of establishing the superiority of the proposed method over existing morphology based methods, the results of the proposed method are compared with that of the method proposed by [4], this method is purposefully chosen, as it is claimed to be the best method in the literature. The results of the [4] method is given in Fig-4 (c) for the input images shown in Fig-4 (a). It can be observed from the Fig-4, that the method proposed by [4] has detected many spurious corner points. On the other hand the proposed method has not detected even a single spurious corner point. Fig-3 and Fig-5 show the results of the proposed method for the shapes in different orientations and scaling factors, which reveals the invariance property of the proposed method in detecting corner points. The objects shown in Fig-5 contains moderate amount of noise, the results of the proposed method on these objects demonstrates the noise withstanding property of the proposed method.

4 Discussion

The major problem with the existing boundary based corner detection schemes is that they are less efficient since at every boundary point, it is required to compute the region of support and as well the curvature at that point to decide if the point is a corner point. In addition they are parametric in nature. The advantage of these methods is that all true corner points are detected. On the other hand, the morphological operation based approaches appear to be more efficient since they selectively decide the parts of the boundary which have corner points to locate the true corner points. But they are less effective as they failed to detect all true corner points. Unlike these, the proposed method being simple approach is both efficient and effective. In addition, the proposed method is nonparametric, as it does not require any input parameter either to decide the size or shape of the structuring element. The existing morphology based methods ([5], [2], [1]) demand input parameters to decide the size and shape of the structuring element and the performance of these methods are heavily driven by the chosen input parameters.

5 Conclusion

A simple method for corner detection using mathematical morphology is presented in this paper. The method is based on morphological skeleton. Corner points are the points where the morphological skeleton intersects the boundary of the image. The proposed method unlike the other morphology based methods detects all corner points present in the extracted corner clusters. In addition the proposed method is nonparametric. The experimental results reveal that the proposed method outperforms the existing morphology based corner detection schemes.

References

1. Laganiere R, (1998). A morphological operator for corner detection, Pattern Recognition, Vol. 31, No. 11, pp. 1643-1652.
2. Lin R.S, Chyi-Hwa Chu and Yuang-Cheh Hsue, (1998). A modified morphological corner detector. Pattern Recognition Letters Vol. 19, pp. 279-286.
3. Shih F.Y, C.F Chuang and V. Gaddipati, (2005). A modified regulated morphological corner detector, Pattern Recognition Letters vol. 26, No. 7, pp. 931-937.
4. Yu L.W, L Hua and Z.G Xi, (2001). A fast algorithm for corner detection using the morphologic skeleton, Pattern Recognition Letters, Vol. 22, pp. 891-900.
5. Zhang X and D. Zhao, (1997). A parallel algorithm for detecting dominant points on multiple digital curves. Pattern Recognition, Vol. 30. No. 2, pp. 239-244.

Correspondence Search in the Presence of Specular Highlights Using Specular-Free Two-Band Images

Kuk-Jin Yoon and In-So Kweon

Robotics and Computer Vision Lab.,
Dept. Electrical Engineering and Computer Science, KAIST, Korea
{kjyoon, iskweon}@kaist.ac.kr

Abstract. In this paper, we present a new method to deal with specular highlights in correspondence search. The proposed method is essentially based on the specular-free two-band image that we introduce to deal with specular reflection. For given input images, specular-free two-band images are generated using simple pixel-wise computations in real-time. Specular-free two-band images are then used to compute per-pixel raw matching costs. By using the specular-free two-band images instead of input images, reliable raw matching costs that are independent of the specularities of image pixels are obtained. As a result, we can find correct correspondences even in the presence of specular highlights. Experimental results show that the proposed method successfully produces accurate disparity maps for stereo images with specular highlights.

1 Introduction

Correspondence search has been a long lasting research topic in the computer vision community since that is the crux of many classical computer vision problems such as motion estimation, object tracking, object recognition , 3D structure reconstruction, etc. To solve the correspondence problem, many methods have been proposed for last decades [1, 2, 3, 4, 5, 6] (See [7] for more information). Most correspondence search methods first compute per-pixel raw matching costs using pixel intensities or colors to measure the similarity between image pixels, assuming that the surfaces in a scene are perfectly Lambertian so that the ICA (Intensity Conservation Assumption) is valid in all input images. However, unfortunately, specular highlights due to non-Lambertian surfaces are frequent in real situations. Because specular reflection makes the intensities and the colors of corresponding pixels different according to the viewpoints, the ICA is not valid any more and the per-pixel raw matching costs for the pixels in specular highlights are erroneous. As a result, severe matching errors occur in specular highlights when using existent correspondence search methods.

Nevertheless, there is a relatively small amount of work to deal with specular highlights in correspondence search. To prevent the errors due to specular highlights, Bhat and Nayar [8] analyzed the physics of specular reflection and the geometry of stereopsis which lead to a relationship between stereo vergence, surface roughness, and the likelihood of a correct match. Based on this analysis, an optimal binocular stereo configuration is determined, which maximizes precision in depth estimation despite specular reflection. Zickler et al. [9] presented a new method that is named as the Helmholtz stereopsis to overcome the specular reflection problem. In this method, stereo images are

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 761–770, 2006.

obtained by switching the positions of a light source and a camera to prevent the color changes due to specular highlights. However, these two approaches require specialized camera configurations, which make the methods impractical. Some other methods tried to deal with specular highlights just by using given input images. In [10, 11], specular pixels in multi-view images are detected first by computing the uncertainty of depth estimates. Detected pixels are then treated as outliers when computing the similarity between pixels to reduce the effect of specular reflection. Yang et al. [13] proposed a new photo-consistency measure that is valid for both diffuse and specular surfaces based on the observation that the reflected colors for most surfaces are co-linear in the RGB color space. However, these methods need many input images to detect specular pixels and to estimate the disparities of specular pixels. Some methods that can be applied to the two-frame stereo problem also have been proposed focusing on the similarity computation. Kim et al. [12] proposed an EM(energy minimization)- and MI(mutual information)-based method without assuming that scene points have similar intensities in different views. The key contribution of their work is to develop the data term that uses mutual information. We also presented an adaptive support-weight method that can deal with specular reflection by adjusting support-weights in the similarity computation step [6]. However, this method requires an accurate analysis of specular reflection to compute support-weights.

In this work, we propose a new method that can efficiently deal with specular highlights in correspondence search. The proposed method is essentially based on the specular-free two-band image that we introduce to deal with specular reflection. It provides a specularity-invariant image representation that can be used for many computer vision methods such as shape from shading and reflection components separation as well as correspondence search. Throughout this work, we assume that the images are taken by cameras with the gamma correction off and that all input images and pixels are chromatic and that there is no saturated pixel.

2 Reflection Model and Image Formation

To deal with specular highlights, we model the image formation process by using the dichromatic reflection model that describes both diffuse and specular reflection.

There are two kinds of reflection under the dichromatic reflection model: diffuse and specular. Diffuse reflection is caused by the subsurface scattering of light, and specular reflection is caused by the surface reflection, as with a mirror. The dichromatic reflection model for dielectric materials, which was proposed by Shafer [16], suggests that the spectral factor can be expressed as the linear weighted sum of two reflectance functions. When an image is taken by a camera, image formation can be described as

$$\begin{bmatrix} I_r(\mathbf{x}) \\ I_g(\mathbf{x}) \\ I_b(\mathbf{x}) \end{bmatrix} = m_d(\mathbf{x}) \begin{bmatrix} \Lambda_r(\mathbf{x}) \\ \Lambda_g(\mathbf{x}) \\ \Lambda_b(\mathbf{x}) \end{bmatrix} + m_s(\mathbf{x}) \begin{bmatrix} \Gamma_r(\mathbf{x}) \\ \Gamma_g(\mathbf{x}) \\ \Gamma_b(\mathbf{x}) \end{bmatrix} \tag{1}$$

$\mathbf{\Lambda}$ = $[\Lambda_r(\mathbf{x}), \Lambda_g(\mathbf{x}), \Lambda_b(\mathbf{x})]^{\mathrm{T}}$ denotes the diffuse chromaticity at $\mathbf{x}$ and $\mathbf{\Gamma}(\mathbf{x})$=$[\Gamma_r(\mathbf{x}), \Gamma_g(\mathbf{x}), \Gamma_b(\mathbf{x})]^{\mathrm{T}}$ the specular or illuminant chromaticity at $\mathbf{x}$. $m_d(\mathbf{x})$ and $m_s(\mathbf{x})$ are the diffuse and specular reflection coefficients, which depend on scene geometry at $\mathbf{x}$. To summarize, the first term and the second term of the right side in Eq. (1) represent the diffuse reflection component and the specular reflection component, respectively.

In this work, it is assumed that input images are taken under the white illumination so that the color of specular reflection is pure-white regardless of an image position in input images. If input images are taken under the non-white illumination, we normalize the input images by using illuminant colors. When assuming a uniform illuminant color in a scene, the illuminant color for a given image can be estimated by using existing color constancy methods [18, 19, 20, 21, 22]. Once the illuminant color is estimated, we can normalize the input image using the estimated illuminant color to yield an image that has pure-white specular components as if it were taken under the white illumination. In this case, Eq. (1) can be simply rewritten as

$$\begin{bmatrix} I_r(\mathbf{x}) \\ I_g(\mathbf{x}) \\ I_b(\mathbf{x}) \end{bmatrix} = m_d(\mathbf{x}) \begin{bmatrix} \Lambda_r(\mathbf{x}) \\ \Lambda_g(\mathbf{x}) \\ \Lambda_b(\mathbf{x}) \end{bmatrix} + m_s(\mathbf{x}) \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} . \tag{2}$$

3 Errors in Raw Matching Costs Due to Specular Reflection

Because support aggregation and correspondence selection are performed by using raw matching costs, it is important to get errorless raw matching costs. However, when one of corresponding pixels have a large specular reflection component, the computed raw matching cost tend to be erroneous because specular reflection makes the intensities and the colors of corresponding pixels different according to the viewpoints.

Suppose that the pixel at $\mathbf{x}$ in the reference image corresponds to the pixel at $\mathbf{x}'$ in the target image. Because the diffuse reflection is independent of viewing directions, two pixels have the same diffuse color (i.e. $[\Lambda_r(\mathbf{x}), \Lambda_g(\mathbf{x}), \Lambda_b(\mathbf{x})]$ = $[\Lambda_r(\mathbf{x}'), \Lambda_g(\mathbf{x}'), \Lambda_b(\mathbf{x}')]$) and, therefore, the color difference between those pixels can be expressed as

$$\begin{bmatrix} I_r(\mathbf{x}) \\ I_g(\mathbf{x}) \\ I_b(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} I_r(\mathbf{x}') \\ I_g(\mathbf{x}') \\ I_b(\mathbf{x}') \end{bmatrix} = \big(m_d(\mathbf{x}) - m_d(\mathbf{x}')\big) \begin{bmatrix} \Lambda_r(\mathbf{x}) \\ \Lambda_g(\mathbf{x}) \\ \Lambda_b(\mathbf{x}) \end{bmatrix} + \big(m_s(\mathbf{x}) - m_s(\mathbf{x}')\big) \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \tag{3}$$

Here, $m_d(\mathbf{x}) = m_d(\mathbf{x}')$ (i.e., $\big(m_d(\mathbf{x}) - m_d(\mathbf{x}')\big) = 0$) because diffuse reflection coefficients are independent of viewing directions. Therefore, Eq. (3) is simplified as

$$\begin{bmatrix} I_r(\mathbf{x}) \\ I_g(\mathbf{x}) \\ I_b(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} I_r(\mathbf{x}') \\ I_g(\mathbf{x}') \\ I_b(\mathbf{x}') \end{bmatrix} = \big(m_s(\mathbf{x}) - m_s(\mathbf{x}')\big) \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \tag{4}$$

When the raw matching cost between two pixels $e(\mathbf{x}, \mathbf{x}')$ is computed in the AD (Absolute Differences) manner, it is expressed as

$$e(\mathbf{x}, \mathbf{x}') = \sum_{c \in \{r,g,b\}} |I_c(\mathbf{x}) - I_c(\mathbf{x}')| = |m_s(\mathbf{x}) - m_s(\mathbf{x}')| \tag{5}$$

From Eq. (5), it is clear that, although two pixels are corresponding to each other, their raw matching cost may not be zero when the scene surface that two pixels come from is not perfectly Lambertian. For that reason, raw matching costs computed by using pixel colors tend to be erroneous for non-Lambertian surfaces and this causes severe matching errors in specular highlights.

4 Error Reduction in Specular Highlights

According to the observation described in the previous section, we have to find out some specularity invariance for each pixel, which is independent of the specularity (i.e. specular reflection coefficient) of a pixel, to get correct correspondences regardless of specular reflection. For this, we propose a specular-free two-band image that provides a specularity-invariant image representation.

4.1 Specularity-Invariant Representation: Specular-Free Two-Band Image

A specular-free image is a specularity-invariant representation of an input image, which is free from specular reflection and has the same geometrical profile as the diffuse reflection component of the input image. Recently, a few methods have been proposed to generate a specular-free image [14, 23, 24, 15]. The resultant specular-free images are used for reflection components separation [14, 15] and surface shape recovery [23, 24]. In this work, we propose a new method for specular-free image generation that is more intuitive, faster, and also proper to correspondence search.

The idea of the proposed specular-free two-band image is shown in Fig. 1. Two dimensional representation is given for visualization. Suppose that we have two adjacent pixels at $\mathbf{x}_1$ and $\mathbf{x}_2$ with the same diffuse color. When denoting the diffuse and specular reflection components of two pixels as $\mathbf{R}_d(\mathbf{x}_1)$, $\mathbf{R}_d(\mathbf{x}_2)$, $\mathbf{R}_s(\mathbf{x}_1)$, and $\mathbf{R}_s(\mathbf{x}_2)$, respectively, the pixel intensities at $\mathbf{x}_1$ and $\mathbf{x}_2$ can be expressed as

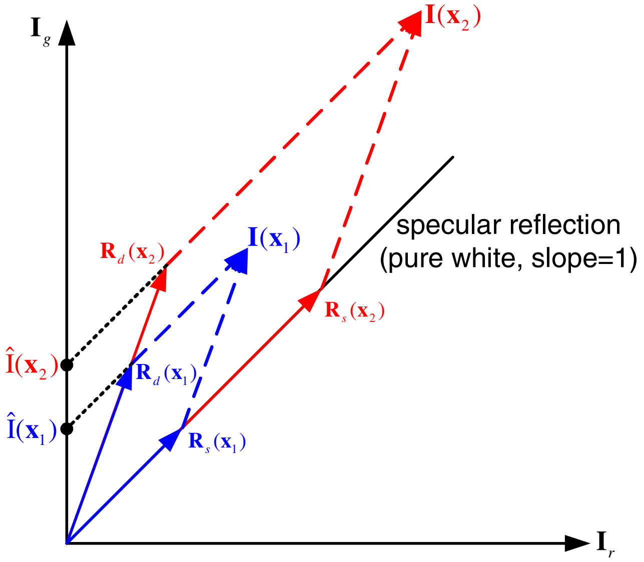

$\mathbf{I}_g$
$\mathbf{I}(\mathbf{x}_2)$
$\mathbf{I}(\mathbf{x}_1)$
$\mathbf{R}_d(\mathbf{x}_2)$
$\mathbf{R}_d(\mathbf{x}_1)$
$\mathbf{R}_s(\mathbf{x}_2)$
$\mathbf{R}_s(\mathbf{x}_1)$
specular reflection
(pure white, slope=1)
$\hat{\mathbf{I}}(\mathbf{x}_2)$
$\hat{\mathbf{I}}(\mathbf{x}_1)$
$\mathbf{I}_r$

Fig. 1. Specularity-invariant value and ratio. All $\hat{I}$ values are specularity-invariant and the local ratios of them provide the geometric profile of diffuse reflection components.

$$\mathbf{I}(\mathbf{x}_1) = \mathbf{R}_d(\mathbf{x}_1) + \mathbf{R}_s(\mathbf{x}_1) \tag{6}$$

$$\mathbf{I}(\mathbf{x}_2) = \mathbf{R}_d(\mathbf{x}_2) + \mathbf{R}_s(\mathbf{x}_2) \tag{7}$$

Here, when $I_g(\mathbf{x}_1) \geq I_r(\mathbf{x}_1)$ and $I_g(\mathbf{x}_2) \geq I_r(\mathbf{x}_2)$, we can compute $\hat{I}(\mathbf{x}_1)$ and $\hat{I}(\mathbf{x}_2)$ simply as

$$\hat{I}(\mathbf{x}_1) = I_g(\mathbf{x}_1) - I_r(\mathbf{x}_1)\ ,\ \hat{I}(\mathbf{x}_2) = I_g(\mathbf{x}_2) - I_r(\mathbf{x}_2) \tag{8}$$

since the color of specular reflection is pure-white.

From Fig. 1, we can see that $\hat{I}$ is independent of the specular reflection component and depends only on the diffuse reflection component — $\hat{I}$ is specularity-invariant. In addition, the following ratio is dependent on only diffuse reflection components.

$$\parallel \hat{I}(\mathbf{x}_1) \parallel : \parallel \hat{I}(\mathbf{x}_2) \parallel = \parallel \mathbf{R}_d(\mathbf{x}_1) \parallel : \parallel \mathbf{R}_d(\mathbf{x}_2) \parallel \tag{9}$$

Based on this observation, we propose a new method to generate a specular-free image.

Let $\tilde{I}(\mathbf{x}) = \min\{I_r(\mathbf{x}), I_g(\mathbf{x}), I_b(\mathbf{x})\}$ and $\tilde{\Lambda}(\mathbf{x}) = \min\{\Lambda_r(\mathbf{x}), \Lambda_g(\mathbf{x}), \Lambda_b(\mathbf{x})\}$. Then, the relationship between $\tilde{I}(\mathbf{x})$ and $\tilde{\Lambda}(\mathbf{x})$ is easily derived from Eq. (2) as

$$\tilde{I}(\mathbf{x}) = \min\{I_r(\mathbf{x}), I_g(\mathbf{x}), I_b(\mathbf{x})\} = m_d(\mathbf{x}) \times \tilde{\Lambda}(\mathbf{x}) + \frac{1}{3} m_s(\mathbf{x}) \tag{10}$$

Since $\tilde{I}(\mathbf{x})$ can be computed simply, we can get the following values for each pixel.

$$\begin{bmatrix} \hat{I}_r(\mathbf{x}) \\ \hat{I}_g(\mathbf{x}) \\ \hat{I}_b(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} I_r(\mathbf{x}) \\ I_g(\mathbf{x}) \\ I_b(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} \tilde{I}(\mathbf{x}) \\ \tilde{I}(\mathbf{x}) \\ \tilde{I}(\mathbf{x}) \end{bmatrix} = m_d(\mathbf{x}) \begin{bmatrix} \Lambda_r(\mathbf{x}) - \tilde{\Lambda}(\mathbf{x}) \\ \Lambda_g(\mathbf{x}) - \tilde{\Lambda}(\mathbf{x}) \\ \Lambda_b(\mathbf{x}) - \tilde{\Lambda}(\mathbf{x}) \end{bmatrix} \tag{11}$$

As shown in Eq. (11), $\hat{I}_r(\mathbf{x}), \hat{I}_g(\mathbf{x})$, and $\hat{I}_b(\mathbf{x})$ are independent of the specular reflection coefficient $m_s(\mathbf{x})$. In addition, they have the same geometrical profile as the diffuse reflection component of the input image. Therefore, a specular-free image is simply generated by subtracting $\tilde{I}(\mathbf{x})$ from the intensities in all color bands as Eq. (11). This is named as the specular-free two-band image because one of $\hat{I}_r(\mathbf{x}), \hat{I}_g(\mathbf{x})$, and $\hat{I}_b(\mathbf{x})$ is zero according to the definition of $\tilde{I}(\mathbf{x})$ and $\tilde{\Lambda}(\mathbf{x})$.

The proposed method for specular-free two-band image generation is very simple. An input image can be transformed into a specular-free two-band image in real-time. In fact, the specular-free two-band image generation can be achieved by a one-line MATLAB instruction as shown in Algorithm 1.

Algorithm 1. MATLAB code for specular-free image generation

```
I: RGB input image
ISF: specular-free two-band image
ISF=I-repmat(min(I,[ ],3), [1,1,3])
```

4.2 Raw Matching Cost Using Specular-Free Two-Band Images

The proposed specular-free two-band images can be efficiently used for correspondence search in the presence of specular highlights as well as reflection component separation and shape recoverty. Suppose again that the pixel at $\mathbf{x}$ in the reference image corresponds to the pixel at $\mathbf{x}'$ in the target image. Then, $\tilde{\Lambda}(\mathbf{x}) = \tilde{\Lambda}(\mathbf{x}')$ because two pixels have the same diffuse color (i.e. $[\Lambda_r(\mathbf{x}), \Lambda_g(\mathbf{x}), \Lambda_b(\mathbf{x})] = [\Lambda_r(\mathbf{x}'), \Lambda_g(\mathbf{x}'), \Lambda_b(\mathbf{x}')]$). The color difference between these two pixels in the specular-free two-band images is then expressed as

$$\begin{bmatrix} \hat{I}_r(\mathbf{x}) \\ \hat{I}_g(\mathbf{x}) \\ \hat{I}_b(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} \hat{I}_r(\mathbf{x}') \\ \hat{I}_g(\mathbf{x}') \\ \hat{I}_b(\mathbf{x}') \end{bmatrix} = (m_d(\mathbf{x}) - m_d(\mathbf{x}')) \begin{bmatrix} \Lambda_r(\mathbf{x}) - \tilde{\Lambda}(\mathbf{x}) \\ \Lambda_g(\mathbf{x}) - \tilde{\Lambda}(\mathbf{x}) \\ \Lambda_b(\mathbf{x}) - \tilde{\Lambda}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

Equation (12) is very obvious because $\hat{I}_c(\mathbf{x})$ and $\hat{I}_c(\mathbf{x}')$ values in specular-free two-band images are independent of specular reflection and $m_d(\mathbf{x}) = m_d(\mathbf{x}')$. Therefore, the raw matching cost between corresponding pixels is equal to zero. Therefore, matching errors due to specular highlights can be greatly reduced by using specular-free two-band images instead of input images.

5 Experiments

The proposed method was applied to synthetic and real images with specular highlights. All real images used for experiments are obtained by cameras with the gamma correction off.

The specular-free two-band images for the images with specular highlights are shown in Fig. 2. The specular-free two-band images for relatively simple images are shown in the first two rows while the specular-free two-band images for highly textured images are shown in the last two rows. We can see that, although the color of each pixel is changed during the transformation, the specularity of each pixel is close to zero in a specular-free two-band image. In addition, we can see that the shading information of an input image is correctly preserved in a specular-free two-band image. Because the transformation from an input image to a specular-free two-band image is achieved by pixel-wise local operations, a highly textured image with specular highlights can be also transformed into a specular-free two-band image without any difficulties.

We then tried to produce dense disparity maps for the rectified synthetic and real stereo images with specular highlights by using the specular-free two-band images. For correspondence search, we have used some different correspondence search methods such as the simple SAD-based method, the adaptive support-weights method [6], and the dynamic programming method [2]. In fact, the proposed specular-free two-band image can be easily used for any existent correspondence methods to make the correspondence method robust to specular reflection.

The correspondence search results for some images with specular highlights are shown in Figs. 3 – 6. Each figure shows input images (left and right) and specular-free two-band images with the resultant disparity maps when using input images and specular-free two-band images, respectively. When using input images, severe matching errors occur in the areas corresponding to specular highlights in both input images regardless of correspondence search methods. However, as we expected, severe matching errors due to specular highlights are greatly reduced by using specular-free two-band images instead of input images.

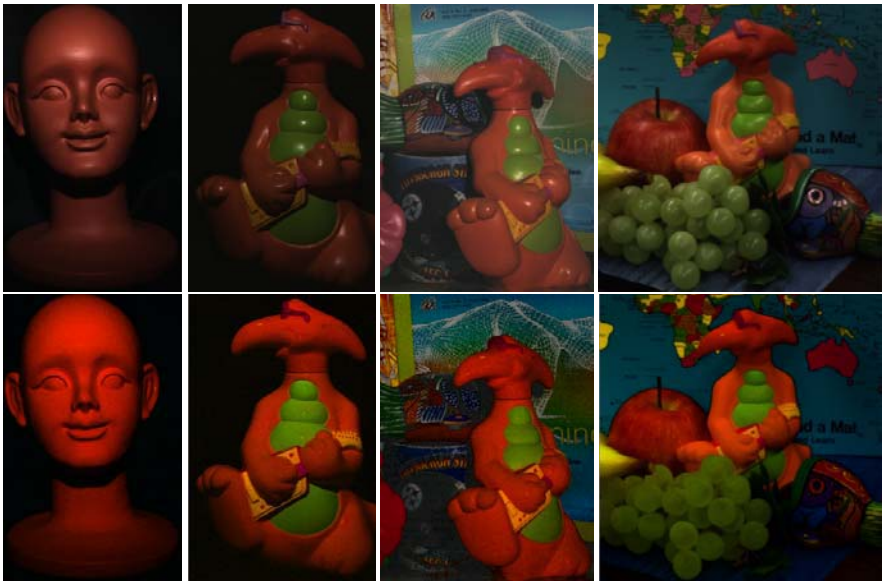
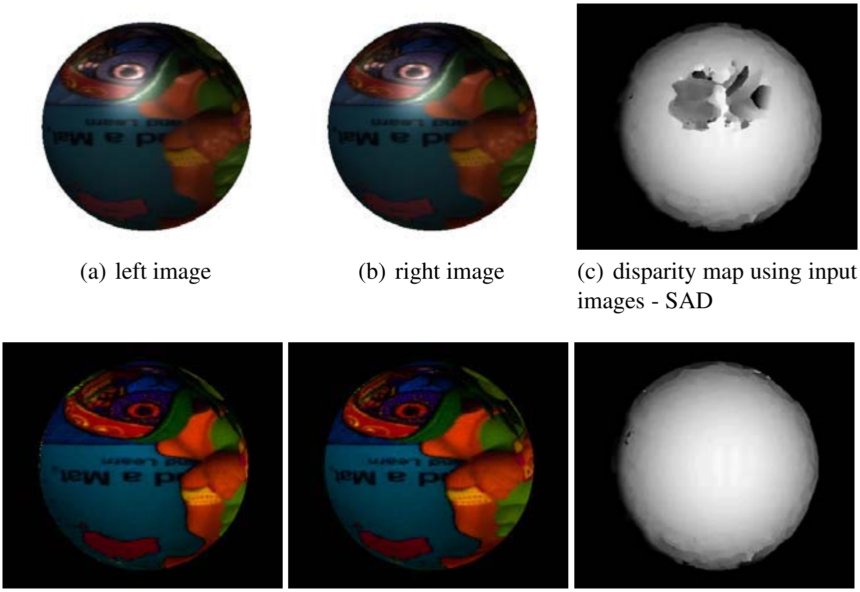


Fig. 2. Results of specular-free two-band image generation. Note that the shading information is preserved in specular-free two-band images.



(a) left image

(b) right image

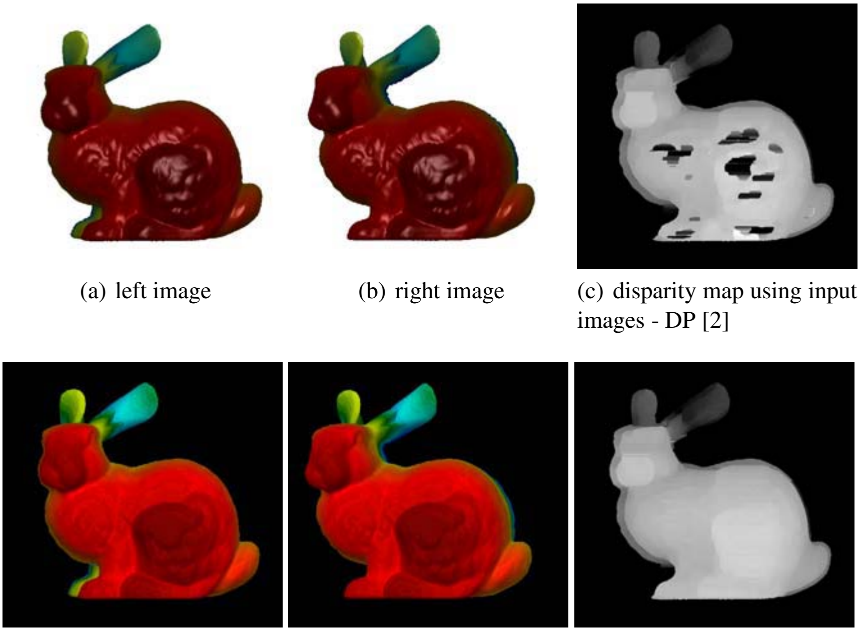
(c) disparity map using input images - SAD

(d) specular-free two-band image of a left image

(e) specular-free two-band image of a right image

(f) disparity map using specular-free two-band images - SAD

Fig. 3. Correspondence search results for synthetic images with specular highlights (1)



(a) left image

(b) right image

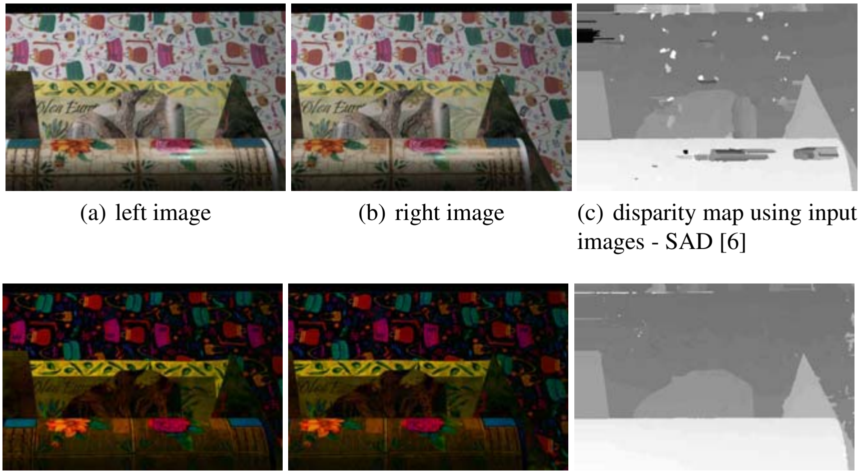
(c) disparity map using input images - DP [2]

(d) specular-free two-band image of a left image

(e) specular-free two-band image of a right image

(f) disparity map using specular-free two-band images - DP [2]

Fig. 4. Correspondence search results for synthetic images with specular highlights (2)



(a) left image

(b) right image

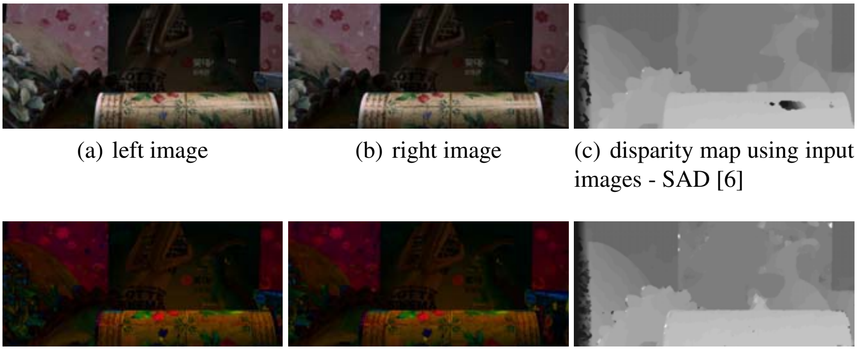
(c) disparity map using input images - SAD [6]

(d) specular-free two-band image of a left image

(e) specular-free two-band image of a right image

(f) disparity map using specular-free two-band images - SAD [6]

Fig. 5. Correspondence search results for real images with specular highlights (1)



(a) left image

(b) right image

(c) disparity map using input images - SAD [6]

(d) specular-free two-band image of a left image

(e) specular-free two-band image of a right image

(f) disparity map using specular-free two-band images - SAD [6]

Fig. 6. Correspondence search results for real images with specular highlights (2)

6 Conclusion

In this work, we have proposed a new method for correspondence search in the presence of specular highlights. For given input images, we first generate specular-free two-band images and measure the similarity between pixels by using specular-free two-band images. By using the specular-free two-band images instead of input images, we get reliable raw matching costs that are independent of the specularities of image pixels. As a result, we can find correct correspondences even in the presence of specular highlights.

The proposed method is essentially based on the specular-free two-band image. It provides a specularity-invariant image representation that can be used for many computer vision problems such as reflection component separation and shape from shading as well as correspondence search. The propose method for specular-free image generation is very simple, fast, and proper to correspondence search. Input images can be transformed into specular-free two-band images in real-time so that the proposed method can be applied to real-time applications. In addition, the proposed specular-free two-band image can be easily used for any existent correspondence methods to make the correspondence methods robust to specular reflection.

Acknowledgments

We would like to thank Dr. Robby T. Tan in NICTA for providing color test images.

References

1. Kanade, T., Okutomi, M.: A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiments, IEEE Trans. Pattern Analysis and Machine Intelligence, Volumn 16 (1994) 920–932
2. Bobick, A. F., Intille, S. S.: Large Occlusion Stereo, Int'l J. Computer Vision, Volumn 33 (1999) 181–200

3. Kang, S. B., Szeliski, R., Jinxjang, C.: Handling Occlusions in Dense Multi-View Stereo, Proc. IEEE Conf. Computer Vision and Pattern Recognition, Volumn 1 (2001) 103–110
4. Kolmogorov, V., Zabih, R.: Computing Visual Correspondence with Occlusions using Graph Cuts, Proc. Int'l Conf. Computer Vision, Volumn 2 (2001) 508–515
5. Veksler, O.: Stereo Correspondence with Compact Windows via Minimum Ratio Cycle, IEEE Trans. Pattern Analysis and Machine Intelligence, Volumn 24 (2002) 1654–1660
6. Yoon, K. -J., Kweon, I. -S.: Locally Adaptive Support-Weight Approach for Visual Correspondence Search, Proc. IEEE Conf. Computer Vision and Pattern Recognition, Volumn 2 (2005) 924–931
7. Scharstein, D., Szeliski, R.: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithm, Int'l J. Computer Vision, Volumn 47 (2002) 7–42
8. Bhat, D. N., Nayar, S. K.: Stereo and Specular Reflection, Int'l J. Computer Vision, Volumn 26 (1998) 91–106
9. Zickler, T. E., Belhumeur, P. N., Kriegman, D. J.: Helmholtz Stereopsis: Exploiting Reciprocity for Surface Reconstruction, Int'l J. Computer Vision, Volumn 49 (2002) 215–227
10. Li, Y., Lin, S., Lu, H., Kang, S. B., Shum, H. -Y.: Multibaseline Stereo in the Presence of Specular Reflections, Proc. Int'l Conf. Pattern Recognetion, (2002) 573–576
11. Lin, S., Li, Y., Kang, S. B., Tong, X., Shum, H. -Y.: Diffuse-Specular Separation and Depth Recovery from Image Sequences, Proc. European Conf. Computer Vision, (2002) 210–224
12. Kim, J., Kolmogorov, V., Zabih, R.: Visual Correspondence Using Energy Minimization and Mutual Information, Proc. Int'l Conf. Computer Vision, (2003) 1033–1040
13. Yang, R., Pollefeys, M., Welch, G.: Dealing with Textureless Regions and Specular Highlights — A Progressive Space Carving Scheme Using a Novel Photo-consistency Measure," in *Proc. Int'l Conf. Computer Vision*, pp. 576–584, 2003.
14. Tan, R. T., Ikeuchi, K.: Separating Reflection Components of Textured Surfaces using a Single Image, Int'l Conf. Computer Vision, (2003) 870–877
15. Tan, R. T., Ikeuchi, K.: Reflection Components Decomposition of Textured Surfaces using Linear Basis Functions, Proc. IEEE Conf. Computer Vision and Pattern Recognition, Volumn 1 (2005) 125–131
16. Shafer, S.: Using Color to Separate Reflection Components,Color Res. Appl., Volumn 10 (1985) 210–218
17. Lee, H. -C., Breneman, E. J., Schulte, C. P.: Modeling Light Reflection for Computer Color Vision, IEEE Trans. Pattern Analysis and Machine Intelligence, Volumn 12 (1990) 402–409
18. Tan, R. T., Nishono, K., Ikeuchi, K.: Color Constancy Through Inverse-Intensity Chromaticity Space, J. Opt. Soc. Amer. A (JOSA A), Volumn 21 (2004) 321–334
19. Klinker, G. J., Shafer, S. A., Kanade, T.: A Physical Approach to Color Image Understanding, Int'l J. Computer Vision, Volumn 4 (1990) 7–38
20. Finlayson, G. D., Schaefer, G.: Solving for Color Constancy Using a Constrained Dichromatic Reflection Model, Int'l J. Computer Vision, Volumn 42 (2001) 127–144
21. Lee, H. C.: Method for Computing the Scene-Illuminant Chromaticity from Specular Highlights, J. Opt. Soc. Am. A, Volumn 3 (1986) 29–33
22. Lehmann, T. M., Palm, C.: Color Line Search for Illuminant Estimation in Real-World Scenes, J. Opt. Soc. Am. A, Volumn 18 (2001) 2679–2691
23. Miyazaki, D., Tan, R. T., Hara, K., Ikeuchi, K.: Polarization-based Inverse Rendering from a Single View, Proc. Int'l Conf. Computer Vision, (2003) 982–987
24. Mallick, S., Zickler, T., Kriegman, D., Belhumeur, P.: Beyond Lambert: Reconstructing Specular Surfaces Using Color, Proc. IEEE Conf. Computer Vision and Pattern Recognition, Volumn 2 (2005) 619–626

Stereo Matching Algorithm Using a Weighted Average of Costs Aggregated by Various Window Sizes

Kan'ya Sasaki, Seiji Kameda, and Atsushi Iwata

Graduate School of Advanced Sciences of Matter, Hiroshima University,
1-3-1 Kagamiyama, Higashi-Hiroshima, 739-8530 Japan
{kanya, kameda, iwa}@dsl.hiroshima-u.ac.jp

Abstract. A window-based stereo matching, which matches pixel values within a window between two images, produces a dense disparity map, and as a result, constructs a dense depth structure. Many algorithms of the window-based stereo matching have been proposed. The conventional algorithms, however, face a trade-off between accuracies of the disparity map in disparity continuity and discontinuity regions due to the window size dependence. In this paper, to solve the issue, we proposed a new algorithm of the window-based stereo matching. In the algorithm, the disparity map is computed using a weighted average of costs aggregated by various window sizes from large to small. Therefore, our algorithm improves accuracy of the disparity map in both disparity continuity and discontinuity regions. In order to evaluate the performance, we have designed C++ programs. The simulation result shows that our algorithm is effective compared to conventional algorithms.

1 Introduction

Stereo matching produces a dense disparity map by using a pair of left and right images of a stereo camera system, and as a result, constructs a dense 3-dimensional depth structure. The stereo matching algorithm is categorized into three major groups: phase-based[1], feature-based[2] and intensity-based[3]. The phase-based algorithm uses phase values of two stereo images processed by spatial band-pass filters to compute a disparity map. The algorithm has advantage of DC independence by low cut filter, but disadvantage of phase-wraparound. The feature-based algorithm uses extracted features from the two images, such as edge, straight line and curve, etc. The algorithm realizes high-speed processing, but cannot generate a dense disparity map due to sparse features. The intensity-based algorithm uses pixel intensities in the two images. Generally, the intensity-based algorithm can generate a dense disparity map while the processing speed is inferior to that of the feature-based algorithm. Therefore, the intensity-based algorithm can be applied to a view synthesis and an image-based rendering that are remarkable applications, which require a dense depth map.

In the intensity-based algorithm, a window-based and a coarse-to-fine algorithms are commonly known as typical approaches. The window-based algorithm

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 771–780, 2006.

matches intensity values within windows between two stereo images. The conventional window-based algorithms, however, face a trade-off between accuracies of the disparity map in disparity continuity and discontinuity regions due to the window size dependence. To solve the problem, the coarse-to-fine algorithm has been proposed[4],[5]. The coarse-to-fine algorithm starts the matching process by the largest window size and gradually decreases the window size with narrowing a range of candidates. However, the conventional coarse-to-fine algorithm often cannot find true disparities due to a limitation of the range of candidates. In this paper, to solve these issues, we proposed a new algorithm of the window-based stereo matching.

2 Intensity-Based Stereo Correspondence Algorithm

A fundamental process of window-based algorithm is generally divided into four steps; matching cost computation, cost aggregation, disparity computation and disparity refinement. Many algorithms have been proposed in each step[3]. The processing flow is explained below. The first step is a matching cost computation. The matching cost means a similarity between left and right pixel intensities in two stereo images. There are some matching cost computation methods: absolute intensity differences (AD), squared intensity differences (SD), cross-correlation, and etc. For example, the matching cost of the AD, C_{mat}, is defined as

$$C_{mat}(x, y, d) = |I_r(x, y) - I_m(x + d, y)|. \tag{1}$$

Here, in the two stereo images, one is a reference image, the other is a matching image. $I_r(x, y)$ is a pixel intensity at (x, y) in the reference image. And $I_m(x + d, y)$ is a intensity of pixel shifted in horizontal by the disparity value, d, from (x, y) in the matching image. Therefore, if d is true disparity, the matching cost, $C_{mat}(x, y, d)$, is reduced to almost zero because I_r is approximately equal to I_m at the disparity. The matching cost is computed for every pixel position to form a matching cost map. The matching cost map is computed for every possible disparity value.

In the next step, the matching costs within a window are aggregated. Because the cost aggregated within the window (hereinafter called an aggregated cost) allows a comparison of texture and inhibition of noise component, there is a clear difference in the aggregated cost between true and fault disparities. The cost aggregation has been implemented using box, binominal, Gaussian filters and shiftable window, etc[6],[7],[8],[9]. If the box filter is used for the cost aggregation, the aggregated cost, $C_{agg}(x, y, d)$, is defined as

$$C_{agg}(x, y, d) = \frac{1}{i \times j} \sum_{i,j \in W} C_{mat}(x + i, y + j, d), \tag{2}$$

where, W denotes the window region. Thus, the matching costs, C_{mat}, within the window, W, are aggregated by moving average at a given disparity, d. The aggregated cost map is computed for every matching cost map.

In the disparity computation step, the best disparity is selected by comparing the aggregated costs across all disparities. Various disparity computation methods have been proposed: local optimization, global optimization and cooperative algorithm, etc. In these methods, the most simple and widely used disparity computation method is a winner-take-all (WTA) optimization categorized the local optimization[3]. The WTA finds a disparity when the aggregated cost is minimum value at each pixel position. The disparity of the minimum aggregated cost is defined as the best disparity. And the disparity selected for every pixel position forms the disparity map. A processing time of the WTA is faster than that of the global optimization.

In the last step, the sub-pixel disparity refinement is computed by fitting a curve to the aggregated costs at discrete pixel units to increase a resolution of the disparity map[3],[10]. It is note that the intensities being matched must vary smoothly to compute accurately[3].

3 Issues of the Conventional Algorithms

In the conventional window-based stereo algorithms, the optimal window size depends on variation in disparity value around a given pixel position. The dependence of the window size is explained below by using Fig. 1. Fig. 1 shows a disparity map of a box. The disparity map is divided roughly into two regions, disparity continuity region, A, and discontinuity region, B. The disparity continuity region, A, is defined as a region where the all disparities are same. Fig. 2(a), (b) show aggregated costs in the disparity continuity region using a large window and a small window at the true disparity when a noise is injected. We assume that the AD and the box filter are used as the matching cost computation and the cost aggregation, respectively. As shown in Fig. 2(a) and (b), matching costs around a given pixel are almost zero because the disparity is true. But, an adjacent matching cost at a given pixel is 20 by noise component. In the case, the aggregated cost is reduced to almost zero as the window size increased because of a smoothing of the noise component. Thus, in the disparity continuity region, a larger window is desirable to avoid the noise influence.

In contrast, the disparity discontinuity region, B, is defined as a region where some disparities are existed. Fig. 2(c), (d) show aggregated costs in the disparity

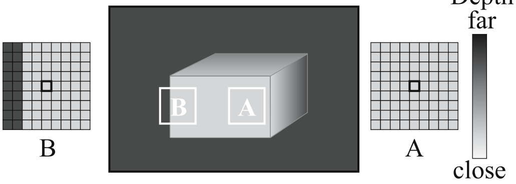


Fig. 1. Disparity map of a box, A: disparity continuity region, B: disparity discontinuity region

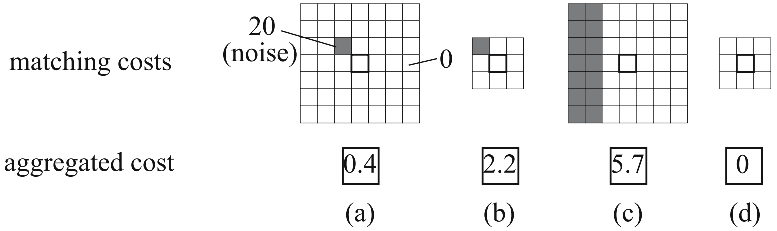


Fig. 2. Aggregated cost computation in a disparity continuity region with a large window (a), with a small window (b), and in a disparity discontinuity region with a large window (c), with a small window (d)

discontinuity region using a large window and a small window at the true disparity where there are different disparities on the left side at a given pixel. As shown in Fig. 2(c), the aggregated cost using the large window becomes large because there are two different disparities within the window. However, in case of the small window, because there is only one disparity in the window, the aggregated cost can be reduced to zero though the given pixel approaches an edge between the different disparities. Thus, in the disparity discontinuity region, a small window is desirable to avoid including the different disparities.

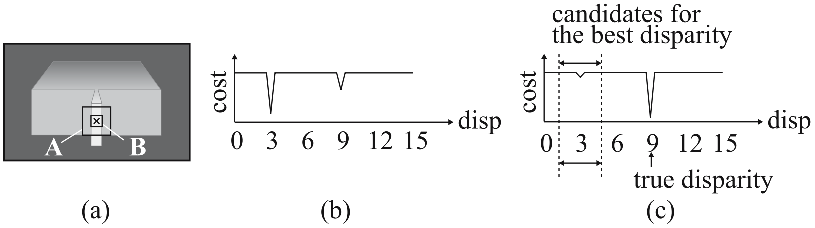


Fig. 3. Issue of the coarse-to-fine algorithm, (a) a disparity map of a box with plated-like bulge, (b) the aggregated cost using a large window “A” against disparity, (c) aggregated cost using a small window “B”

The coarse-to-fine algorithm solves the issue by using multiple costs aggregated by various window sizes. However, the conventional coarse-to-fine algorithm is difficult to find a true disparity of small object differed vastly from a disparity of background. Fig. 3(a) shows a disparity map of a box with plate-like bulge at the front. The aggregated cost is computed at a center of the bulge by the coarse-to-fine algorithm. Fig. 3(b) shows the aggregated costs using a large window, A, against disparity. The aggregated cost at a disparity of the background is smaller than at a disparity of the bulge due to a strong influence of the background. Thus, a disparity of background, 3, is selected as the best disparity. And then the aggregated costs using a small window, B, as shown in Fig. 3(c). In this case, the aggregated cost at the disparity of the bulge is smaller than at a disparity of the background. In the coarse-to-fine algorithm, however, candidates for the best disparity are limited within a given area centered on the disparity computed by the large window. Therefore, the disparity of the bulge is not selected as the best disparity.

4 Proposed Algorithm

To solve above issues, we propose a new window-based and coarse-to-fine like stereo matching algorithm. A process flow of the proposed algorithm is almost the same as the conventional process flow in the section 2. The AD, which is given by Equ. (1), and the Gaussian filter are used as the matching cost computation and the cost aggregation, respectively. The aggregated cost by the Gaussian filter, $C_{agg}(x, y, d)$, is given by

$$G(i,j) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{i^2+j^2}{2\sigma^2}\right), \tag{3}$$

$$C_{agg}(x,y,d) = \sum_{i,j} G(i,j) C_{mat}(x+i, y+j, d), \tag{4}$$

where, σ^2 is a variance of the Gaussian distribution and the filter size increased with the σ. The Gaussian filter has a better performance than the box filter in the disparity discontinuity region because a weight of the Gaussian filter is the largest at a given pixel position and decreases with distance from the pixel position.

In the proposed algorithm, the aggregated cost maps are computed in sequential order while the window size is reduced gradually, like the coarse-to-fine algorithm. And a new cost (hereinafter called an averaged cost) map is computed using a weighted average of the aggregated cost maps recursively and given by

$$C[n] = \begin{cases} C_{agg}[n], & n = 1, \\ \dfrac{w_1 \cdot C[n-1] + w_2 \cdot C_{agg}[n]}{w_1 + w_2}, & n \geq 2. \end{cases} \tag{5}$$

Here, $C[n]$ and $C_{agg}[n]$ are the averaged and aggregated costs in n-th iteration, respectively. And w_1 and w_2 are weights of the averaged and aggregated costs, respectively. At first, an aggregated cost, $C_{agg}[1]$, is computed using the largest window for every possible disparity value at each pixel and is equal to the averaged cost in first iteration, $C[1]$. Then, a next aggregated cost, $C_{agg}[2]$, is computed at each pixel using a window whose size is reduced compared to the first iteration. The averaged cost, $C[2]$, is renewed at each pixel using the weighted average of the present aggregated cost, $C_{agg}[2]$, and the previous averaged cost, $C[1]$, according to the Equ. (5) for every possible disparity. These processes are computed recursively while the window size is reduced gradually. The final averaged cost map, $C[N]$, is computed when the window becomes the minimum size and has every characteristic of aggregated costs using various window sizes.

In the next step, the WTA optimization is used as the disparity computation. The WTA finds a disparity, d, when the averaged cost, $C[N](x, y, d)$, is minimum value at each pixel, (x, y). And the disparity map is formed by the disparity of the minimum averaged cost. In the last step, a parabolic approximation is used as the sub-pixel disparity refinement. The parabola fits three values that are the averaged costs at the selected disparity by the WTA and both adjacent disparities.

In the proposed algorithm, because the aggregated costs computed for every possible disparity value, the issue of limitation of candidates in the coarse-to-fine algorithm is solved. And, to construct a dense disparity map, the weight of the aggregated cost increases with reducing the window size due to the recursive formula (5). The characteristic of the averaged cost is changed by the maximum and minimum sizes of the window, a reduction ratio of the window size and the increasing rate of the weight, which is controlled by a ratio between w_1 and w_2 and an iteration count of the recursive formula, N.

5 Simulation

We have designed C++ programs of the proposed algorithm in order to evaluate the performance compared with the other conventional algorithms. We used a stereo image data, *Tsukuba*, from the Middlebury stereo evaluation page[11] for our simulation as shown in Fig. 4. Fig. 4(a) and (b) show a reference image, which is one of the stereo images, and a true disparity map (16 scales) , respectively. We have computed disparity maps according to the proposed algorithm in the section 4. The iteration count of the recursive formula (5) was five and the window size was gradually reduced, $\sigma = 24, 12, 6, 3, 1.5$. And both of the weights, w_1 and w_2, were set to one. Fig. 5(a) shows the disparity map computed by the averaged cost at first iteration. The disparity map was broadly correct compared with the true disparity map and inhibited the noise component significantly though detailed characteristics of objects in the disparity discontinuity region could not be detected because the computation at the first iteration used only the largest window. Fig. 5(b) shows the final disparity map. The detailed characteristics in the disparity discontinuity region, such as poles and edges of the lump, were detected and there was a little terrible error in the disparity continuity region since the final averaged cost computed by Equ. (5) contained every characteristic of aggregated costs using various window sizes from large to small. To explain the effect of the proposed algorithm, Fig. 6 shows costs for disparity in disparity

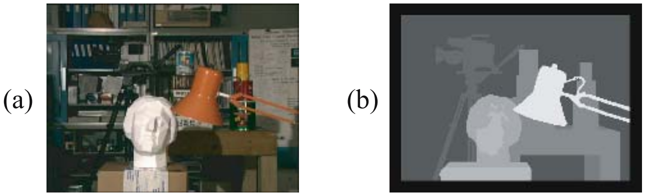


Fig. 4. Simulation images: (a) reference image, (b) true disparity map

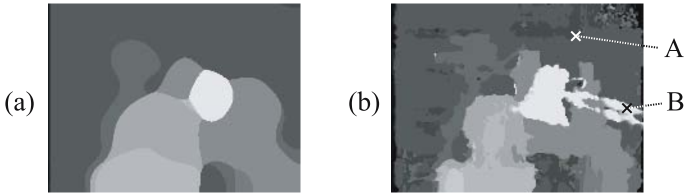


Fig. 5. Disparity maps by our algorithm at the first iteration (a) and at the final iteration (b)

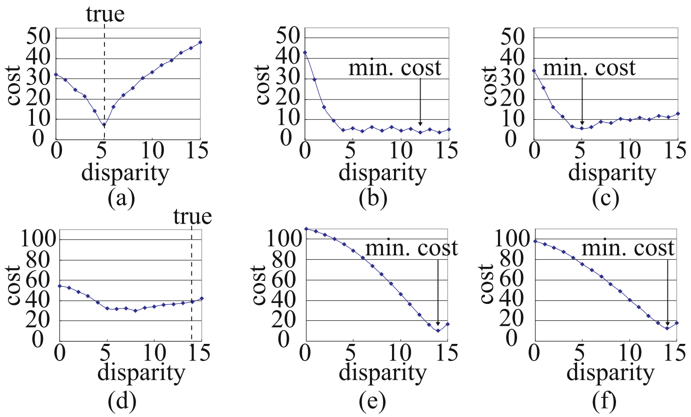


Fig. 6. Comparison of aggregated costs with the averaged cost in disparity continuity region, aggregated costs using a large window (a) and a small window (b), the final averaged cost (c), and in disparity disconitnuity region, aggregated costs using a large window (d) and a small window (e), the final averaged cost (f)

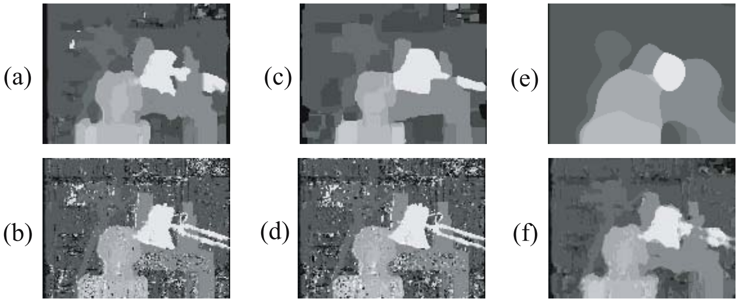


Fig. 7. Disparity maps: (a) box filter (window size = 15), (b) box filter (window size = 3), (c) shiftable window (window size = 21), (d) shiftable window (window size = 3), (e) coarse-to-fine (the first iteration), (f) coarse-to-fine (the final iteration)

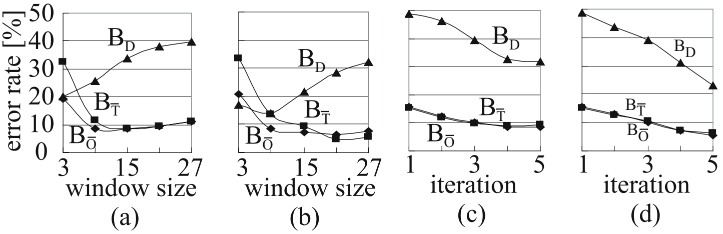


Fig. 8. Plots of the three evaluation measures of (a) box filter, (b) shiftable window, (c) coarse-to-fine algorithm and (d) our algorithm

continuity and discontinuity regions at pixel positions A and B in Fig. 5(b). In these figures, (a) and (b) show the aggregated costs using the largest and smallest window size against disparity in the disparity continuity region, respectively, and (c) shows the final averaged cost. As shown in Fig. 6(c), in the disparity continuity region, the averaged cost at the true disparity was minimum due to a influence of the aggregated cost using the largest window though there was a little difference in the aggregated cost using the smallest window around the true disparity and the true disparity was not selected. By the same token, as shown in Fig. 6(f), in the disparity discontinuity region, the averaged cost at the true disparity was minimum due to a influence of the aggregated cost using the smallest window though the aggregated cost using the largest window was not minimum. Therefore, the proposed algorithm computed the disparity map well in both of the disparity continuity and discontinuity regions. And, as shown in Fig. 6, the aggregated cost using the smallest window had more strongly influence to the averaged cost than that using the largest window to detect detail characteristics of objects. We compared our algorithm with the conventional algorithms. We selected the box filter and the shiftable window for the cost aggregation as the conventional window-based algorithms. The box filter, which is introduced in the section 2, is simple and therefore widely used. The shiftable window is introduced as the best aggregation method in the review [3]. Additionally, the conventional coarse-to-fine algorithm was simulated to the comparison. To compare the cost aggregation step, these conventional algorithms used the same methods as the proposed algorithm in the other steps. Namely, the AD, the WTA optimization and the parabolic approximation are used as the matching cost computation, the disparity computation and the sub-pixel refinement, respectively. Fig. 7(a) and (b) show disparity maps by the box filter using the large and small windows, respectively. And Fig. 7(c) and (d) show disparity maps by the shiftable window using the large and small windows, respectively. The sizes of the shiftable window were selected to get the best results. As shown in Fig. 7(a) and (c), when these algorithms used the large box filter and shiftable window, these algorithms had similar characteristics to the proposed algorithm at first iteration (Fig. 5(a)). However, as shown in Fig. 7(b) and (d), when these algorithms used the small box filter and shiftable window, the disparity maps of these algorithms had many terrible errors though the detailed characteristics in the disparity discontinuity region were detected.

The coarse-to-fine algorithm used the same filter and condition as the proposed algorithm. Namely, the iteration count was five and the window size of the Gaussian filter was gradually reduced, $\sigma = 24, 12, 6, 3, 1.5$. In the coarse-to-fine algorithm, the range of candidate was reduced, $\pm 4, 3, 2, 1$, with reducing the window size. Fig. 7(e) shows the disparity map by the coarse-to-fine algorithm at first iteration. The first disparity map of the coarse-to-fine algorithm was exactly the same as that of the proposed algorithm because of the same condition. Fig. 7(f) shows the final disparity map by the coarse-to-fine algorithm. As shown in Fig. 7(f), the coarse-to-fine algorithm had almost similar characteristics to the proposed algorithm and there was a little terrible error in the disparity continuity region. However, the detailed characteristics in the disparity discontinuity region, such as poles and edges of the lump, were not detected due to an influence of the background and this result indicated the issue of the coarse-to-fine algorithm in the section 3. Therefore, the coarse-to-fine algorithm has the limitation to detect the detailed characteristics. Fig. 8(a), (b), (c) and (d) show plots of the three evaluation measures, $B_{\bar{O}}$, $B_{\bar{T}}$ and B_D, of the box filter, the shiftable window, the coarse-to-fine algorithm and the proposed algorithm, respectively. $B_{\bar{O}}$ is the error rate in the non-occluded region. $B_{\bar{T}}$ is the error rate in the texture-less region. The texture-less region includes a part of the disparity continuity region since disparities within the texture-less region are same unless this region is slanted. B_D is the error rate in the disparity discontinuity region. The error rate represents the percentage of bad pixels, which mean false disparities compared with the true disparities as shown in Fig. 4(b). In the Fig. 8, the horizontal axis measures window size and the vertical axis measures the error rates. As shown in Fig. 8(a) and (b), in the plots of the box filter and the shiftable window, when window size is small, $B_{\bar{T}}$ is high and B_D is low. In contrast, when window size is large, $B_{\bar{T}}$ is low and B_D is high. Namely, these results indicate the issue of the trade-off accuracies of the disparity map in disparity continuity and discontinuity regions against the window size. As shown in Fig. 8(c) and (d), in the coarse-to-fine and proposed algorithms, $B_{\bar{T}}$ and B_D decrease with increasing the iteration. However, in the coarse-to-fine algorithm, B_D does not decrease in the final iteration because of the limitation of candidates. Contrastively, in the proposed algorithm, B_D decreases more than the coarse-to-fine algorithm. Therefore, these above results indicate that the proposed algorithm solves the issue of the conventional window-based and coarse-to-fine algorithms in the section 3. The table 1 shows the best results of the evaluation measures, $B_{\bar{O}}$, $B_{\bar{T}}$, B_D and B_A, in the conventional and proposed

Table 1. The best results in the conventional and proposed algorithms

	$B_{\bar{O}}$[%]	$B_{\bar{T}}$[%]	B_D[%]	B_A[%]
box filter	8.40	8.65	33.85	10.20
shiftable win.	6.31	4.64	28.36	8.06
coarse-to-fine	8.48	9.23	31.59	10.31
our algorithm	5.57	6.20	23.14	7.64

algorithms. B_A is the error rate in all regions. As shown in the table 1, B_A of the proposed algorithm is minimum. Therefore the proposed algorithm got better results than the conventional algorithms.

6 Conclusion

We proposed a new window-based and coarse-to-fine like stereo matching algorithm. The disparity map was computed using a weighted average of costs aggregated by various window sizes from large to small. And we have designed C++ programs to evaluate the performance. The proposed algorithm solved the issue of the trade-off between accuracies of the disparity map in disparity continuity and discontinuity regions against the window size, and the limitation of candidates of the coarse-to-fine algorithm.

References

1. A. Cozzi et al, "Performance of phase-based algorithms for disparity estimation," *Machine Vision and Application* Vol. 9, Issue 7, pages 334-340, 1996.
2. F. Candocia, M. Adjouadi, "A Similarity Measure for Stereo Feature Matching," *IEEE Trans. on Image Processing*, Vol. 6, No. 10, pages 1460-1464, 2003.
3. D. Scharstein and R.Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *IJCV* 47(1/2/3):7-42, April-June 2002.
4. A. Witkin, D. Terzopoulos, and M. Kass, "Signal matching through scale space," *IJCV*, 1:133-144, 1987.
5. K. J. Hanna and Neil E. Okamato, "Combining stereo and motion analysis for direct estimation of scene structure," *in Proc. Intl. Conf. on Computer Vision*, pp. 357-365, 1993.
6. R. Kimura et al, "A convolver-based real-time stereo machine (SAZAN)," *CVPR*, volume 1, pages 457-463, 1999.
7. P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications* COM-31(4):532-540, 1983.
8. D. Scharstein and R. Szeliski, "Stereo matching with nonlinear diffusion," *IJCV*, 28(2):155-174, 1998.
9. H. Tao, H. Sawhney, and R. Kumar, "A global matching framework for stereo computation," *ICCV*, volume I, pages 532-539, 2001.
10. Q. Tian and M. N. Huhns, "Algorithms for subpixel registration," *CVGIP*, 35:220-233, 1986.
11. D. Scharstein and R.Szeliski, "Middlebury Stereo Vision Page," *www.middlebury.edu/stereo.*

Pseudo Measurement Based Multiple Model Approach for Robust Player Tracking

Xiaopin Zhong, Nanning Zheng, and Jianru Xue

Institute of Artificial Intelligence and Robotics,
Xi'an JiaoTong University, Xi'an, China
{xpzhong, nnzheng, jrxue}@aiar.xjtu.edu.cn

Abstract. This paper presents a robust player tracking method for sports video analysis. In order to track agile player stably and robustly, we employ multiple models method, with a mean shift procedure corresponding to each model for player localization. Furthermore, we define pseudo measurement via fusing the measurements obtained by mean shift procedure. And the fusing coefficients are built from two likelihood functions: one is image based likelihood; the other is motion based association probability. Experimental results show effectiveness of our method in the hard case of player tracking literature.

1 Introduction and Related Work

Object tracking is one of crucial communities in computer vision. Good solutions to this problem (i.e. Real time and robust tracking) have a variety of applications such as navigation [6][8], missile defense [6], surveillance, human computer interface [7], intelligent transportation system and so on. Its application to sports domain also provides us with individuals moving analysis of team sports [4].

Usually, object tracking consists of two major components: Target Representation and Filtering [5]. In radar tracking domain, target is a simple echo and is only represented by coordinate value. So here filtering, which aims at target dynamics, is more important than target representation model. Whereas in visual tracking literature, target is large enough to be modeled by its appearance, shape, color or other specific features. When the sample rate of image data sequence is high enough, motion of the visual target between two consecutive frames will be negligible. Thereby, target representation is more important than filtering techniques in visual tracking literature. However, we pay attention to both of target representation and filtering because of the character of player tracking.

In sports video analysis, it's quite difficult to track players stably since they are highly non-rigid, identical dressing, dynamic uncertainty and occlusion of teammates frequently occurs. The method based on template matching [9] is easy to drift; therefore it is inevitable to be inaccurate in tracking and easy to loose the track. [10] and [11] used the top view to track handball game players indoor. [12] tracked multiple players in a video of American football. [13] used boosting

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 781–790, 2006.

technique to reinforce the proposal distribution of particle filter for tracking multiple hockey players. [14] tracked athletes via multiple features of multiple views. They all pay attention to only one point of view, target representation or filtering. There are a lot of research works on how to remove camera motion from a game video [2], so it is out of scope in this paper and we use a static camera here.

Even though the accuracy is less important to help the audience to enjoy the sports, filtering do much help to tackle the difficulties of player tracking, such as uncertainty of dynamics of agile player, partial occlusion of two or more players and clutter background. We insist on a good motion model so much that multiple model approach for hybrid system [8] will be chosen to track players. Consequently, a new algorithm, *pseudo measurement based multiple model*, is proposed for player tracking. It employs multiple model method, with a mean shift procedure corresponding to each model for player localization. Pseudo measurement is built via linear fusion technique by two likelihood function: one is image based likelihood; the other is motion based association probability. An important motivation for this idea is cue integration between image and motion to overcome the weakness of individual cue. Hence, pseudo measurement based multiple model algorithm is adaptive to some hard problem in player tracking literature, such as non-rigid target and agile motion.

We begin in section 2 with player localization. In section 3, the proposed pseudo measurement based multiple model method is introduced. Experimental results and some minor problems present in section 4. Finally, conclusion and future work are discussed in section 5.

2 Mean Shift Based Player Localization

In this section, we first recall the well-known mean shift procedure for player localization, and then discuss the hard situations in localization of players.

Mean shift is a nonparametric estimator of density gradient. When used in computer vision, color based mean shift is robust and also fast [5][15].

Color based mean shift models the target by the color histogram. Let $\{x_i\}_{i=1}^n$ be a set of n points in $\mathbb{R}^2$ space to represent pixel locations of target. Then the probability of color u in the target model is derived by employing a convex and monotonic decreasing kernel function $k : [0, \infty) \to R$.

[5] has defined a distance metric,

$$d(y) = (1 - \sum_u \sqrt{\hat{p}_u(y)\hat{q}_u})^{1/2}$$

based on Bhattacharyya coefficient to denote how well the candidate and model match. Maximizing this distance (see [5] for details) yields mean shift vector computed with kernel k and its bandwidth h:

$$M_h(y) = \frac{\sum_{i=1}^n x_i k(\frac{y-x}{h})}{\sum_{i=1}^n k(\frac{y-x}{h})} - y \tag{1}$$

[5] recommends Epanechnikov kernel function. After a few iterations, mean shift vector will converge to zero.

Only localization is not sufficient for filtering technique because of the measurement uncertainty. We assume the measurement uncertainty is Gaussian distribution and use three special point's sum of squared differences (SSD) value to approximate the Gaussian distribution, according to [17].

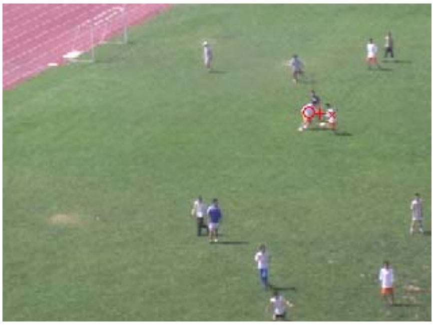


Fig. 1. Three modes (marked in red×, red+ and red∘) are formed nearby two teammates. Consequently localization with mean shift procedure is inaccurate.

Here we notice that only color histogram information used, which will lead to large variations for adjacent location on the image lattice and the spatial information is lost. On the other hand, mean shift algorithm searches a local density extremum. Therefore, mean shift is sensitive to its initial placement. Especially when two players (team mates) get close, we probably obtain error localization with mean shift procedure (e.g. Fig.1). To tackle this difficulty, we use probabilities belonging to the real target, i.e. pseudo measurement presented in section 3, to constrain the localization results.

3 Pseudo Measurement Based IMM Filtering

A principled choice of dynamics of a tracking system is essential for good results. However, players are highly maneuvering targets, which is the reason that leads to awful player track with only one fixed dynamic model. Nowadays, considerable research has been undertaken in the field of hybrid system estimation theory [6] in radar tracking literature. That means we can make use of several dynamic models simultaneously to characterize the target's motion. In our research, we pick IMM (interacting multiple model) method, one of suboptimal filtering techniques, along with a pseudo measurement to fuse multiple models.

In this section, we first introduce pseudo measurement into IMM framework via Bayesian filtering theory, and then rectify the pseudo measurement with additional image based likelihood function and motion based likelihood function. In the end, the whole pseudo measurement based multiple model filtering algorithm we have proposed is listed.

3.1 Pseudo Measurement Based Multiple Model Filtering Framework

In radar tracking literature, IMM has been verified to be a best compromise between optimal filtering and computational complexity [6]. Fig.2 demonstrates our framework.

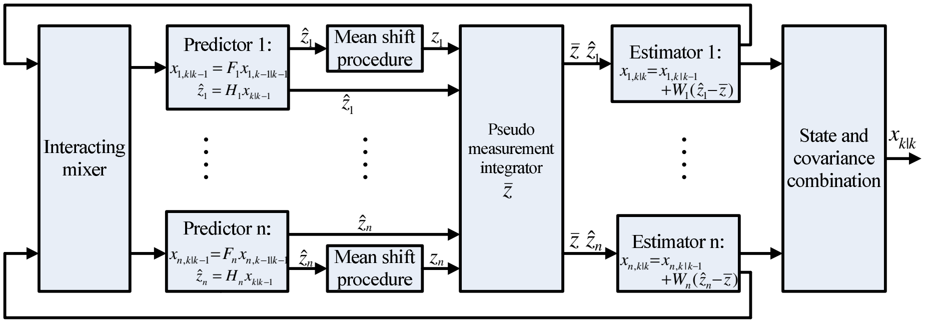

Interacting mixer
Predictor 1:
$x_{1,k|k-1} = F_1 x_{1,k-1|k-1}$
$\hat{z}_1 = H_1 x_{k|k-1}$
$\hat{z}_1$
Mean shift procedure
z_1
$\hat{z}_1$
Predictor n:
$x_{n,k|k-1} = F_n x_{n,k-1|k-1}$
$\hat{z}_n = H_n x_{k|k-1}$
$\hat{z}_n$
$\hat{z}_n$
Mean shift procedure
z_n
Pseudo measurement integrator
$\bar{z}$
$\bar{z}$ $\hat{z}_1$
Estimator 1:
$x_{1,k|k} = x_{1,k|k-1} + W_1(\hat{z}_1 - \bar{z})$
$\bar{z}$ $\hat{z}_n$
Estimator n:
$x_{n,k|k} = x_{n,k|k-1} + W_n(\hat{z}_n - \bar{z})$
State and covariance combination
$x_{k|k}$

Fig. 2. Pseudo Measurement based MM Filtering Framework

According to Fig.2, we define n motion models to form a IMM filter. The *ith* state transition equation and measurement equation are written as

$$\begin{cases} x^i(k+1) = F_i x^i(k) + v_i(k) \\ z^i(k+1) = H_i x(k+1) + w_i(k+1) \end{cases} \tag{2}$$

Where $x^i(k)$ and $z^i(k)$ are state vector and measurement vector belonging to *ith* model at time k. Process noise $v_i(k)$ and measurement noise $w_i(k)$ are independent Gaussian noise with mean zero, covariance $Q_i(k)$ and $R_i(k)$ respectively.

In order to locate the targets (players), a mean shift procedure is employed for each motion model. Then n measurements are produced. Let $\{z_i(k)\}_{i=1}^{m(k)}$ be the measurements at time k, i.e. localizations obtained by mean shift procedure in this paper. However, only one integrated measurement, which is called *pseudo measurement* here, will be used to drive IMM filter. Hence we define pseudo measurement $\bar{z}(k)$ as

$$\bar{z}(k) = \sum_{i=1}^{m(k)} \omega_i(k) \cdot z_i(k) \tag{3}$$

$$\omega_i(k) = \frac{p_i(k)}{\sum_i p_i(k)} \tag{4}$$

Here, $m(k)$ is the number of measurement at time k. And $\omega_i(k)$ is weighting factor determined by the likelihood p_i of each candidate measurement belonging to the real target. The likelihood p_i will be made clear in next subsection. $\bar{z}(k)$ in (3) is named pseudo measurement and is similar with but distinct from probabilistic data association [17], which is a radar tracking fusion strategy to handle the problem of data association when more than one or no measurement emerges. In our situation, each mean shift procedure returns a converging point, thus the number of measurement is changeless, i.e. n. (3) indicates that fused pseudo measurement is more accurate than any individual measurement.

When teammates get close, it's reasonable that motion information is prior to appearance information for player tracking system due to easily confusing the localization of teammates. So we try to employ the prediction of pseudo measurement to emphasize the motion information from multiple motion models. Let $M_j(k)$ be the jth model at time k, then the model probability conditioned on history measurements is

$$p(M_j(k)|Z^{k-1}) = \sum_{i=1}^{n} p(M_j(k)|M_i(k-1), Z^{k-1}) \cdot p(M_i(k-1)|Z^{k-1}) \qquad (5)$$

Where, Z^{k-1} is history measurement up to time $k-1$. $p(M_j(k)|M_i(k-1), Z^{k-1})$ indicates the model transition probability which is preset and $p(M_i(k-1)|Z^{k-1})$ means the previous model probability conditioned on history measurements. For each model, each corresponding filter (such as standard Kalman filter) can calculate a measurement prediction, denoted by $\hat{Z}_j(k)$. Then we achieve the pseudo measurement prediction by

$$\hat{z}(k) = \sum_{j=1}^{n} p(M_j(k)|Z^{k-1}) \cdot \hat{z}_j(k) \qquad (6)$$

This pseudo measurement prediction is crucial in our method in the case of players' occlusion and no real measurement achieved (see next subsection for details).

3.2 Measurement Likelihood

In this subsection, we'll build a straightforward likelihood function for p_i in (4) using appearance information as well as motion information.

In our method, likelihood function of the measurement is defined as below,

$$p_i = (La_i)^{\alpha} \cdot (Lm_i)^{\beta} \qquad (7)$$

Where, La_i denotes the likelihood from target appearance and Lm_i from target motion. α and β are the weights implying the reliabilities of appearance based and motion based information respectively, satisfying $0 \leq \alpha, \beta \leq 1$. (7) indicates the likelihood p_i is more rigorous after considering both points of view in tracking literature: target representation and filtering. In our experiments, we fix α and β for simpleness in spite of their significance for adaptiveness.

Firstly, the image based likelihood La_i function can be many of similarity function, such as image based template matching function, feature based template matching function and even statistics based likelihood function. Without loss of generality and for simpleness, we apply Bhattacharyya coefficient, which has been defined in mean shift procedure [15], to get a robust image based likelihood function. Hence, we define La_i as

$$La_i = \exp(\gamma \cdot \rho_i) \tag{8}$$

$$\rho_i = \sum_l \sqrt{\hat{p}_l(z_i) q_l} \tag{9}$$

Here, ρ_i is Bhattacharyya coefficient between the color distribution of model q and that of candidate measurement $\hat{p}(z_i)$, also an intermediate result from mean shift procedure. Notice that (8) is a nonlinear function and γ is another parameter to adjust the impact of appearance based likelihood. The influence of γ can be grasped more easily in our experiments.

Secondly, when player occlusion occurs, appearance information of target fades out and their motion information should take over the tracker. We assume that the measurement innovation, which is obtained via the pseudo measurement prediction, obeys Gaussian distribution. Similar to IMM's mode likelihood definition, we define Lm_i as

Table 1. Detailed steps of pseudo measurement based MM filtering in one circle

1. Calculate the mixing probabilities: $\mu_{k-1|k-1}(i,j) = \frac{p(i,j)\cdot\mu_{k-1}(i)}{\sum_i p(i,j)\cdot\mu_{k-1}(i)}$
2. Redo the filters' initialization

$$\hat{x}^{(j),0}_{k-1|k-1} = \textstyle\sum_i \hat{x}^{(i)}_{k-1|k-1}\mu_{k-1|k-1}(i,j)$$
$$\nu_{k-1}(i,j) = \hat{x}^{(i)}_{k-1|k-1} - \hat{x}^{(j),0}_{k-1|k-1}$$
$$P^{(j),0}_{k-1|k-1} = \textstyle\sum_i \mu_{k-1|k-1}(i,j)\cdot\{P^{(i)}_{k-1|k-1} + \nu_{k-1}(i,j)\cdot\nu^T_{k-1}(i,j)\}$$

3. Filters' prediction: $\hat{z}_j = H_j \cdot \bar{x}^{(j)}_{k|k-1} = H_j \cdot F_j \cdot \hat{x}^{(j),0}_{k-1|k-1}$
4. Calculate pseudo measurement prediction $\hat{z}(k)$ in (6);
5. Mean shift procedure from $\hat{z}_j$ for player localization z_j and SSD for its uncertainty R_j;
6. Get the appearance likelihood La_i via (8) and (9);
7. Obtain the motion based likelihood Lm_i by (10);
8. Calculate measurement likelihood p_i in (7);
9. Combine pseudo measurement $\bar{z}$ via (3) and (3);
10. All filters run as standard Kalman filter;
11. Update model likelihood and probabilities

$$\Lambda^{(j)}_k = \mathcal{N}\big(\bar{Z} - h(\hat{x}^{(j),0}_{k|k-1}); 0, S^{(j)}_k\big);$$
$$\eta^{(j)}_k = \Lambda^{(j)}_k \textstyle\sum_i p(i,j)\cdot\mu^{(i)}_{k-1}; \quad \mu^{(j)}_k = \frac{\eta^{(j)}_k}{\sum_i \eta^{(i)}_k}$$

12. Estimate and covariance combination

$$\hat{x}_{k|k} = \textstyle\sum_i \hat{x}^{(i)}_{k|k}\mu^{(i)}_k; \quad P_{k|k} = \textstyle\sum_i \mu^{(i)}_k\{P^{(i)}_{k|k} + [\hat{x}^{(i)}_{k|k} - \hat{x}_{k|k}]\cdot[\hat{x}^{(i)}_{k|k} - \hat{x}_{k|k}]^T\}$$

$$Lm_i = \frac{1}{\sqrt{2\pi|S_i|}} \exp\left[- \frac{(z_i - \hat{z})^T \cdot S_i^{-1} \cdot (z_i - \hat{z})}{2} \right] \tag{10}$$

Where $\hat{z}$, the pseudo measurement prediction is introduced in previous subsection and S_i is the innovation covariance which is calculated with measurement covariance R_i in standard Kalman filter. Now the motion based likelihood function Lm_i is indicating that the pseudo measurement is biased to motion prediction, controlled by the parameter α and β.

3.3 Pseudo Measurement Based MM Filtering

In this subsection, the detailed steps of pseudo measurement based MM filtering algorithm for player tracking are present for summary. In Table 1 Some procedures can be achieved from IMM algorithm (seeing [6] for details) directly.

4 Implementation and Results

The proposed method, pseudo measurement based multiple model, has been tested under various football game video. To evaluate the performance of the method, we compared our tracking results with ground truth, marked manually, and with other tracking strategies, such as mean shift and mean shift with Kalman filtering.

4.1 Experiment Configuration

The implementation configuration is set as below. To describe the player state, we use

$$x(k) = [x, v_x, a_x, y, v_y, a_y]_k^T$$

where (x, y) is coordinate of player location in image plane, (v_x, v_y) is its velocity and (a_x, a_y) the acceleration. Since we can only "see" the player's position information in image sequence, our system measurement is denoted by $z_k = [x, y]_k^T$ only. Three models are used to characterize the player motion. They are constant velocity model (CV), constant acceleration model with small noise (LowCA) and constant acceleration model with large noise (HighCA).

Since the football court is looked down, the size of the player varies slightly. Therefore, we won't adapt the model size in our experiments. To be simple, α and β are both set to 1. However, γ adjusts the impact of appearance based likelihood, thereby we set γ to different value to test the effectiveness of our method, seeing next subsection.

4.2 Implementation Results

In this subsection, we test our algorithm with the video sequence "football.avi", compared with other two common algorithms (one is mean shift procedure only,

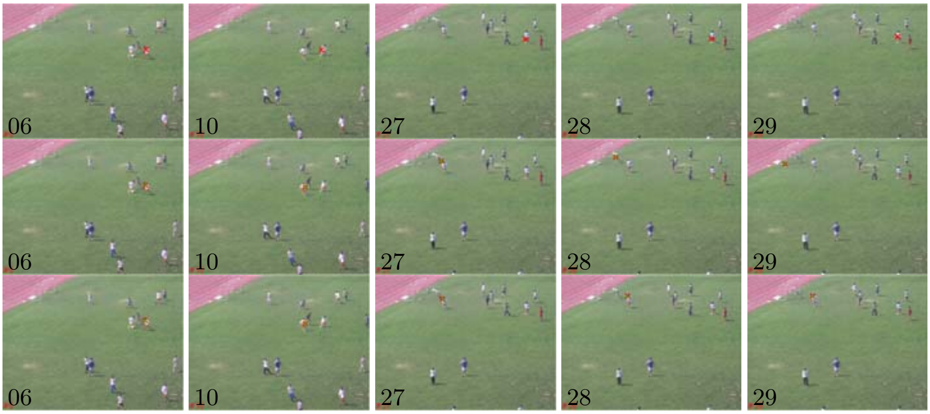


Fig. 3. In these tracking result sequences, player position estimated is marked with a red cross. The first row displays mean shift only tracking result. The second row shows the result of Kalman + mean shift method. And the result of our method, pseudo measurement based multiple model approach, is put in the third row.

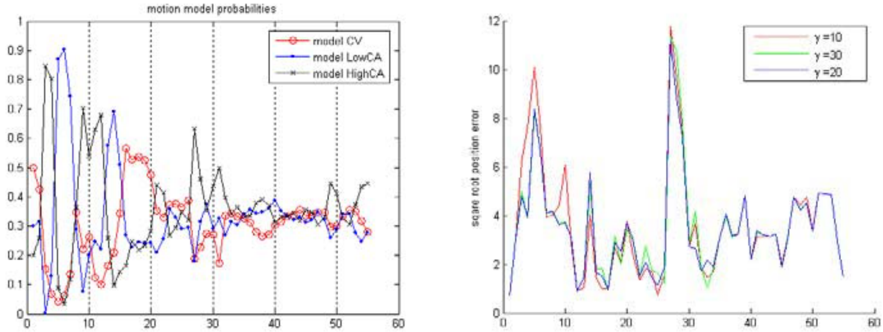


Fig. 4. The left figure is motion model probability of player selected. The right one demonstrates γ adjusting the effectiveness of the image based likelihood.

the other is mean shift with CV based Kalman filtering) in several phases. In video "football.avi" a special target with agile motion is selected to be tracked.

Firstly, the sequences with estimated position marked with Red Cross (Fig.3) are present. In Fig.3, only frames 6\10\27\28\29 are shown as key frames. Obviously, mean shift method failed when two teammates are very close to each other from frame 6 to frame 10, because mean shift can't distinguish them well only by player's appearance. From frame 27 to frame 29 mean shift + Kalman method also failed since the player's position predicted in Kalman filter dropped into the region of another similar player. However, our approach is such a robust tracking method for player tracking that it can succeed in many hard cases. Secondly, the left figure in Fig.4 shows the history of the motion model probabilities for the player selected by our algorithm. Obviously, the motion model probability is not as stable as that in radar literature because the mean shift procedure is not stable for player localization. Thirdly, we redo our method only under the modification of parameter γ, comparing their square root position error with the ground truth marked by hand (the right figure in Fig.4). This experimental result has proven that the image based likelihood did help us to improve the player tracking.

5 Conclusion and Future Works

In this paper, we first present the challenges in player tracking area, for instance, the unknown motion mode and unknown noise level. Then to localize the player, we apply mean shift procedure which has been verified to be robust in visual system. However, mean shift procedure is dependent on the initialization so severely that only one initialization is not enough for robust player tracking. Furthermore, we import a multiple model method designed for hybrid system in radar tracking literature, to get multiple measurements which include true measurement and false measurements. To tackle the multiple measurements problem, a pseudo measurement is designed via two likelihood function: motion based likelihood and image based likelihood.

The experimental results show the performance of our method in player tracking. However, there are several minor problem need to be taken into account further. For example, non-rigid player varying all the time challenges the model of image based likelihood. So a better model updating scheme may help a lot in accuracy. In addition, how to choose the parameter α, β and γ to be adaptive in different cases needs more research.

References

1. Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic Press Inc. (1988)
2. http://vismod.media.mit.edu/vismod/demos/football/tracking.htm
3. Colins, R., Lipton, A., Fujiyoshi, H., Kanade, T.: Algorithms for cooperative multisensor surveillance. Proceedings of the IEEE. Vol. 89, No.10 (2001) 1456–1477
4. http://www.prowess.com.au/infodoc.html
5. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 25, No.5 (2003) 564–577
6. Mazor, E., Averbuch, A., Bar-Shalom, Y., Dayan, J.: Interacting multiple model methods in target tracking: a survey. IEEE Transactions on Aerospace and Electronic Systems. Vol. 34, No. 1 (1998) 103–123
7. Kyung-Nam, K., Ramakrishna, R.S.: Vision-based eye-gaze tracking for human computer interface. IEEE SMC'99 Conference Proceedings. Vol. 2, (1999) 324–329
8. Bar-Shalom, Y., Li, X.R.: Estimation and applications to tracking and navigation: Academic Press Inc. (1995)
9. Seo, Y., Choi, S.H., Kim, H.W., Hong. K.S.: Where are the ball and players? Soccer game analysis with color-based tracking and image mosaic. In Proceedings of International Conference on Image Analysis and Processing, (1997) 196–203

10. Pers, J., Kovacic, S.: Tracking People in Sport: Making Use of Partially Controlled Environment. In Proceedings of the 9^{th} International Conference on Computer Analysis of Images and Patterns, (2001) 374–382
11. Pers, J., Kovacic, S.: Computer Vision System for Tracking Players in Sports Games. In Proceedings of the First Intenatioal Workshop on Image and Signal Processing and Analysis, (2000) 81–86
12. Intille, S., Bobick, A.: Closed-world tracking. In Proceedings of Intenatioal Conference on Computer Vision, (1995) 672–678
13. Okuma, K., Taleghani, A., Freitas, N., Little, J., Lowe, D.: A Boosted Particle filter: Multitarget detection and tracking. In Proceedings of European Conference on Computer Vision 2004, (2004) 28–39
14. Misu, T., Gohshi, S., Izumi, Y., Fujita, Y., Naemura, M.: Robust Tracking of Athletes Using Multiple Features of Multiple Views. In Journal of WSCG, Vol.12, No.1-3 (2004)
15. Comaniciu, D., Ramesh, V.: Mean Shift and Optimal Prediction for Efficient Object Tracking. In Proceedings of the IEEE Intenational Conference on Image Processing, (2000) 70–73
16. Kirubarajan, T., Bar-Shalom, Y., WD Blair, and GA Watson: IMMPDAF for Radar Management and Tracking Benchmark with ECM. IEEE Transaction on Aerospace and Electronic Systems, Vol. 34, No. 4 (1998) 1115–1134
17. Nickls, K., Hutchinson, S.: Estimating Uncertainty in SSD-Based Feature Tracking. Image and Vision Computing, Vol. 20, (2002) 47-58

A Hierarchical Method for 3D Rigid Motion Estimation*

Thitiwan Srinark[1], Chandra Kambhamettu[2], and Maureen Stone[3]

[1] Department of Computer Engineering, Faculty of Engineering,
Kasetsart University, Bangkok 10900, Thailand
thitiwan.s@ku.ac.th

[2] Department of Computer & Information Sciences, University of Delaware,
Newark, DE 19716, USA
chandra@cis.udel.edu

[3] Dept of Biomedical Sciences and Orthodontics, Vocal Tract Visualization Lab,
University of Maryland Dental School,
Baltimore, MD 21201, USA
mstone@umaryland.edu

Abstract. We propose a hierarchical method for 3D rigid motion estimation between two 3D data sets of objects represented by triangular meshes. Multiresolution surfaces are generated from the original surface of each object. These surfaces are decomposed into small patches based on estimated geodesic distance and curvature information. In our method, segment-to-segment matching to recover rigid motions at each resolution level of surfaces is performed. Motion results from low resolution surface matching are propagated to higher resolution surface matching in order to generate a spatial constraint for similar segment selection. Our approach can recover 3D rigid motion of both rigid body and non-rigid body (with partial rigid areas). The method was tested to estimate rigid motions of 3D data obtained by the Cyberware scanner.

1 Introduction

Rigid motion analysis is a fundamental problem in computer vision. It is closely related to the problems of surface matching, surface registration, and object recognition. In the literature, a large number of techniques have been developed to solve these problems. The iterative closest point (ICP) algorithm [1] by Besl and McKay is the most popular method. It has many derivatives improving the original one, e.g., using point-to-normal in the distance evaluation instead of point-to-closest point [2], applying features to match compatible points [3, 4, 5], etc. However, ICP based methods will converge only if good initial approximations are provided, and the closest point searching method does not lead to local minima. In [6], Johnson and Hebert proposed Spin-image a 2D-histogram based

* This work is partially done under National Institutes of Health, Grant No. R01 DC01758.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 791–800, 2006.

representation, that has been successfully applied to surface matching of cluttered objects in 3-D complex scenes. Zhang and Hebert applied Harmonic map in surface matching [7], which emphasizes handling of occlusion and different resolution. Yahia *et al.* used geodesic distance evolution of surfaces in the matching problem [8]. The method can handle a topological change and deformation of surfaces. Sun and Abidi developed point's fingerprint from geodesic circles [9] and used it in image registration, e.g., for matching overlapping surface area.

Most of the previous methods can perform matching of surfaces and partial surfaces, but these surfaces must not be very noisy. Also, good initialization of motion parameters plays essential role in these methods, along with smallness of motion. In our method, flexibility is introduced such that we allow local deformations, missing of some partial surface regions, and noise. Also, no initialization is required. In our method, surfaces of objects are segmented into small patches based on geodesic distance and curvature information. Similarity of segments is measured by a set of error functions which are invariant to rigid motion. We then select patches from each object which are similar to each other to perform matching and estimate possible rigid motions. Some motion results are then filtered out based on error functions which are variant to the rigid motion. The motion that gives the smallest mean distance between surfaces is returned. We apply multiresolution analysis techniques on the original surfaces to generate surfaces that are less complex and contain less details than the original ones. We formulate a hierarchical method for rigid motion estimation such that motion results from coarser levels are propagated to the finer levels.

2 Rigid Motion Estimation

The problem of rigid motion estimation is defined as follows. Given two sets of 3D data points without known correspondence, recover the (global) rigid motion that includes rotation (3-DOF) and translation (3-DOF) between these two data sets. Suppose these two data sets are $\mathbf{O}$ (before motion) and $\mathbf{O}'$ (after motion). We want to find the transformation consisting of the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$ such that $\mathbf{RO} + \mathbf{T} \simeq \mathbf{O}'$. In the following, we present error functions used in our method, and later we explain about our rigid motion estimation algorithm.

2.1 Error Functions

Error functions are defined to measure segment similarity. We have two groups of error functions. The first group consists of error functions that are invariant to rigid transformation such as $\mathsf{EigenDiff}$, $\mathsf{CurvednessDiff}$, and $\mathsf{SizeDiff}$. These three components measure the similarity of two segments and are used in segment selection before applying motions. The second group of error functions are variant to rigid transformation such as $\mathsf{SegDist}$, $\mathsf{NormalDiff}$, $\mathsf{OrientDiff}$, and $\mathsf{ObjDist}$. These error functions measure how well each motion performs in segment-to-segment and object-to-object matching. Suppose S and S' are segments we want to perform matching. S and S' belong to objects $\mathbf{O}$ and $\mathbf{O}'$, respectively. The error functions are defined as follows.

$\mathsf{EigenDiff}$ is the difference of eigenvalues of two covariance matrices. The covariance matrix is computed from (x, y, z) coordinates of vertices in each segment, thus the covariance matrix size is 3×3. Let $\mathbf{A}$ and $\mathbf{A}'$ be covariance matrices of segments S and S', respectively. Let $v_i = (x_i, y_i, z_i)$ be points belonging to S, where $i = 1, ..., n$. To compute the covariance matrix, v_i are centered around $\overline{v_i}$, which is the mean point in S, so $v_i = v_i - \overline{v_i}$. The covariance matrix $\mathbf{A}$ is thus computed as $\mathbf{A} = \mathbf{V}\mathbf{V}^{\mathbf{t}}$. The covariance matrix $\mathbf{A}'$ is also computed in similar fashion. Let $\lambda_1 > \lambda_2 > \lambda_3$ be eigenvalues of $\mathbf{A}/||\mathbf{A}||$; and $\lambda_1' > \lambda_2' > \lambda_3'$ be eigenvalues of $\mathbf{A}'/||\mathbf{A}'||$, where $||.||$ denotes the matrix norm. $\mathsf{EigenDiff}$ is thus defined as,

$$\mathsf{EigenDiff} = \sum_{i=1}^{3} (\lambda_i - \lambda_i')^2 .$$

$\mathsf{CurvednessDiff}$ is the difference of segment curvedness values. The curvedness of a segment is estimated by the curvedness at the segment center. The curvedness at a point is computed by that point's principal curvatures, k_1 and k_2 [10]. The curvedness C is, $C = \sqrt{\frac{k_1^2 + k_2^2}{2}}$. Let C and C' be the curvedness values of segments S and S', respectively. $\mathsf{CurvednessDiff}$ is thus defined as the absolute difference between C and C', $\mathsf{CurvednessDiff} = |C - C'|$.

$\mathsf{SizeDiff}$ is the absolute difference between two segment sizes. The size of a segment is defined using area of the segment. Let $|S|$ and $|S'|$ be the size of S and S', respectively. Then, $\mathsf{SizeDiff} = ||S| - |S'||$.

The next four error functions are applied after rigid motion is recovered. Suppose $(\mathbf{R}, \mathbf{T})$ is one of the recovered motions. Let v_i and n_i be vertices and normals belonging to S, respectively. Thus the motion $(\mathbf{R}, \mathbf{T})$ is applied to S as, $v_i = \mathbf{R}v_i + \mathbf{T}$ and $n_i = \mathbf{R}n_i$.

$\mathsf{NormalDiff}$ is the difference between normals of two segments. Suppose $|S| < |S'|$. Therefore,

$$\mathsf{NormalDiff} = \sum_{i=1}^{|S|} \sqrt{\sum_{k} (n_{ki} - n_{kj})^2} ,$$

where $k \in \{x, y, z\}$, $[n_{xi}, n_{yi}, n_{zi}]$ are normals of the first segment, and $[n_{xj}, n_{yj}, n_{zj}]$ are normals of corresponding vertices in the second segment. The corresponding vertex is estimated from the vertex of the closest match.

$\mathsf{OrientDiff}$ measures how well the surfaces of two objects are aligned with each other after applying motion. Let O_c and O_c' be centroids of objects $\mathbf{O}$ and $\mathbf{O}'$, respectively. The motion $(\mathbf{R}, \mathbf{T})$ is applied to O_c such that $O_c = \mathbf{R}O_c + \mathbf{T}$. Let S_c and S_c' be centroids of segments S and S', respectively. Compute two normals N and N', where $N = (S_c - O_c)/||S_c - O_c||, N' = (S_c' - O_c')/||S_c' - O_c'||$, and $||.||$ denotes the vector norm. Hence $\mathsf{OrientDiff}$ is computed as, $\mathsf{OrientDiff} = 1 - N \cdot N'$. Since N and N' should be aligned with each other in the same direction for good matching, we can discard any motions whose corresponding $\mathsf{OrientDiff} > 1$.

$\mathsf{SegDist}$ and $\mathsf{ObjDist}$ are the average distances between the surfaces of two segments and of two objects, respectively. They are calculated based on computation of distances between surfaces by the GTS library [11]. Surfaces are represented by bounding box trees. The distance from a triangle on a surface to the other surface is computed by (i) sampling points on that triangle, (ii) calculating the distance from each sampling point to the closest object (bounding box) of tree that represents the other object, and (iii) averaging these sampled point distances as the triangle distance. Then the distance between surfaces is defined as the sum of the distances of triangles weighted by their area and divided by the total area of the surface. However, it is possible that $\mathsf{SegDist}(S, S') \neq \mathsf{SegDist}(S', S)$, and $\mathsf{ObjDist}(\mathbf{O}, \mathbf{O}') \neq \mathsf{ObjDist}(\mathbf{O}', \mathbf{O})$. Therefore, we always make sure to compute the distance from the smaller surface to the larger surface such that if $|S| < |S'|$, we compute $\mathsf{SegDist}(S, S')$, otherwise, we compute $\mathsf{SegDist}(S', S)$ (similarly for $\mathsf{ObjDist}(\mathbf{O}, \mathbf{O}')$).

2.2 Rigid Motion Estimation Algorithm

Our rigid motion estimation method consists of six steps: (i) multiresolution analysis, (ii) surface segmentation, (iii) segment selection, (iv) segment-to-segment matching, (v) transformation filtering, and (vi) transformation scoring. The first step generates multiresolution surfaces when the level number is given. The higher number corresponds to the lower resolution. We apply a decimation algorithm [12], which is a fast mesh simplification method, to generate multiresolution surfaces. However, the resulting surfaces are not smooth, making surface analysis difficult. Therefore, we need to apply a surface smoothing method to smooth these simplified surfaces [13]. The second step is to segment surfaces provided by the previous step. This method segments triangular surface meshes based on estimated geodesic distance and curvature. Resulting segments are classified into (1) peak-type, (2) pit-type, (3) minimal surface-type, and (4) flat-type. Each segment type includes vertices with its type, and nearby vertices based on estimated geodesic distance.

In the next step (segment selection), each segment of one surface is compared with each segment of the target surface. From segmentation, segments are classified into different types based on curvatures. Thus, we use this information to select matching pairs such that only segments having the same type are allowed to align with each other. Also we use three error functions that are invariant to rigid transformation to measure similarity between two segments. These error functions are $\mathsf{EigenDiff}$, $\mathsf{CurvednessDiff}$, and $\mathsf{SizeDiff}$. For each segment S_i belonging to $\mathbf{O}$, we compute $\mathsf{EigenDiff}(S_i, S'_j)$, $\mathsf{CurvednessDiff}(S_i, S'_j)$ and $\mathsf{SizeDiff}(S_i, S'_j)$, where S'_j are segments belonging to $\mathbf{O}'$, and $j = 1, ..., m$. Then for each S_i, we compute the average of each error with all S'_j, $\overline{\mathsf{EigenDiff_i}}$, $\overline{\mathsf{CurvednessDiff_i}}$, and $\overline{\mathsf{SizeDiff_i}}$. Therefore, for each S_i, we select S'_j such that $\mathsf{EigenDiff}(S_i, S'_j)$, $\mathsf{CurvednessDiff}(S_i, S'_j)$, and $\mathsf{SizeDiff}(S_i, S'_j)$ are less than $\overline{\mathsf{EigenDiff_i}}$, $\overline{\mathsf{CurvednessDiff_i}}$, and $\overline{\mathsf{SizeDiff_i}}$, respectively.

In segment-to-segment matching, transformations $(\mathbf{R}, \mathbf{T})$ are computed by aligning selected pairs of similar segments together. We compute principal components of vertices in each segment. These principal components are used to form orthogonal matrices and these matrices are used to calculate rigid transformations. Suppose S and S' are segments we want to perform matching where S and S' belong to objects $\mathbf{O}$ and $\mathbf{O}'$, respectively. For each segment, we create the covariance matrix from points in the segment. Eigenvectors and eigenvalues are then computed from the covariance matrix. Eigenvectors are considered as principal components of the segments. Let $\vec{e_i}, \vec{e_j}, \vec{e_k}$ be eigenvectors of S, and let $\vec{e_i}', \vec{e_j}', \vec{e_k}'$ be eigenvectors of S', where $\vec{e_k}$ and $\vec{e_k}'$ are most aligned with the normal directions of S and S', respectively. The orthogonal matrices $\mathbf{E}$ and $\mathbf{E}'$ are formed as follows,

$$\mathbf{E} = [\vec{e_i}, \vec{e_j}, \vec{e_k}],$$
$$\mathbf{E}' = [\vec{e_i}', \vec{e_j}', \vec{e_k}'],$$

where the coordinate system of matrices is defined as,

$$\vec{e_i} \times \vec{e_j} = \vec{e_k}, \vec{e_j} \times \vec{e_k} = \vec{e_i}, \vec{e_k} \times \vec{e_i} = \vec{e_j},$$

$$\vec{e_i}' \times \vec{e_j}' = \vec{e_k}', \vec{e_j}' \times \vec{e_k}' = \vec{e_i}', \vec{e_k}' \times \vec{e_i}' = \vec{e_j}'.$$

There are four combinations of alignment for the two matrices, so $\mathbf{E}'$ is modified by rotating $\vec{e_i}'$ and $\vec{e_j}'$ around $\vec{e_k}'$ by $0°, 90°, 180° and 270°$. Therefore, we have $\mathbf{E_1'} = [\vec{e_i}', \vec{e_j}', \vec{e_k}'], \mathbf{E_2'} = [\vec{e_j}', -\vec{e_i}', \vec{e_k}'], \mathbf{E_3'} = [-\vec{e_i}', -\vec{e_j}', \vec{e_k}'], \mathbf{E_4'} = [-\vec{e_j}', -\vec{e_i}', \vec{e_k}']$. Let $\overline{v}$ and $\overline{v}'$ be mean points of S and S', respectively. Rigid transformations are hence computed as $\mathbf{R_i} = \mathbf{E_i'E^t}, \mathbf{T_i} = \overline{v}' - \mathbf{R_i}\overline{v}$, where $i = 1, 2, 3, 4$. To find the best matching among these four transformations, we apply each (R_i, T_i) to vertices of S and apply R_i to normals of S. Then we use two error functions, SegDist and OrientDiff, to compute how good the alignment of two segments is after applying the motion. We select the one which gives the smallest SegDist and also gives the valid orientation of object alignment, e.g., OrientDiff < 1.

The next step is two-pass transformation filtering. The first pass is to filter transformations within the same group, and the second pass is to filter all transformations selected from the first pass. Suppose the segment S_i of $\mathbf{O}$ is matched with S_j' of $\mathbf{O}'$, where $j = 1, ..., m$ and the transformations $(\mathbf{R_{ij}}, \mathbf{T_{ij}})$ are computed from matching S_i and S_j'. We compute the weighted sum of two error functions, NormalDiff and OrientDiff, to measure matching error in segment level. Let Align be the matching error of two segments,

$$\begin{aligned}\mathsf{Align}(S_i, S_j') = \beta_1 \mathsf{NormalDiff}(S_i, S_j') \\ + \beta_2 \mathsf{OrientDiff}(S_i, S_j'),\end{aligned}$$

where β_1 and β_2 are weighting parameters. In the first pass, the same group of transformations includes all transformations which are computed from the same S_i, i.e., $(\mathbf{R_{ij}}, \mathbf{T_{ij}}), j = 1, ..., m$, are in the same group. For each group, we compute the average of the Align of all $(\mathbf{R_{ij}}, \mathbf{T_{ij}})$, and select only the transformations whose Align are smaller than the average. In the second pass, the new average

Align is computed from Align of all transformations selected from every group. Then only the transformations whose Align are smaller than the new average are passed to the next step.

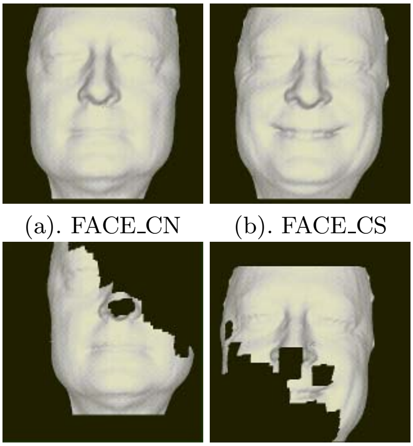
For the remaining transformations, we compute their ObjDist score, and we select the transformation which gives the smallest ObjDist to be the best transformation. The best transformation is then passed to the finer surface matching as an initial transformation. At the finer surface matching, the initial transformation is used as an additional spatial constraint to pair nearby segments in segment selection, thus matching of far away pairs can be cut off. The threshold distance between segments is defined in order to limit the maximum distance between two segments that are allowed to be paired. The size of this threshold is proportional to the size of ObjDist passed from the previous level. The other parts of the method are processed similarly. The method returns the final transformation when the finest surface matching is done, or when ObjDist is below a defined threshold.

3 Experiments

We tested our method on 3D data obtained by the Cyberware scanner; face data is obtained from UIUC, rest of the data is from the www.cyberware.com. The test data consist of two sets: open surface and closed surface objects. The open surface objects include four 3D face data that have different facial expressions and data completeness as shown in Figure 1. The face data are nonrigid objects having some local rigid parts, e.g., nose (nose is elastic, but does not contribute to the nonrigid motion in most facial expressions). We classify the face data experiments into four test groups based on the motion of objects, and completeness of the data surfaces. We have four groups of experiments. In TEST_A and TEST_C, we want to test on pure rigid motion recovery when data surfaces are complete and incomplete, respectively. In TEST_B and TEST_D, we want to test on recovery of rigid motions when nonrigid motion is also present with complete and incomplete data, respectively. Incomplete surfaces mean that only partial data is available. We created the incomplete surfaces using Maya$^{\mathrm{TM}}$, a modeling software package. The closed surface objects include six pairs of objects: the models of Venus head, dinosaur, ball joint, Isis statue, hip, and teeth. Each object is matched with its pair that is locally deformed, added with noise, and whose parts are partially missing by Maya$^{\mathrm{TM}}$.

We applied a set of rigid motions to one of each object pair in both test sets. The set of rigid motions contains seven different transformations $(\mathbf{R}, \mathbf{T})$ with $-135^\circ \leq (\theta_x, \theta_y, \theta_z) \leq 135^\circ$ and $-30 \leq (t_x, t_y, t_z) \leq 30$, where $\theta_x, \theta_y, \theta_z$ are rotation angles about $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$ axes, respectively, and t_x, t_y, t_z are translations along $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$ axes, respectively. We evaluated the results by computing the error distance between the correct rigid motion $(\mathbf{R_c}, \mathbf{T_c})$ and the results $(\mathbf{R_i}, \mathbf{T_i})$. Note that rotations are represented by 3×3 matrices, and translations are represented by 3×1 vectors. The translation error between $\mathbf{T_c}$ and $\mathbf{T_i}$ is computed as

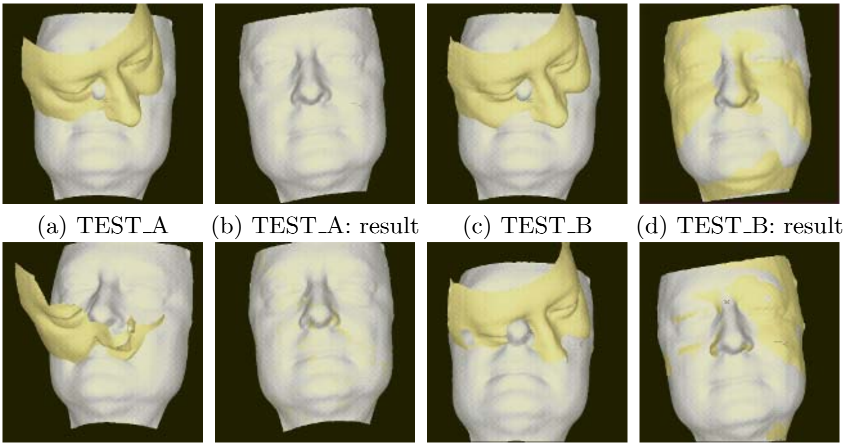
$$\epsilon_t = \sqrt{(\mathbf{T_c} - \mathbf{T_i})(\mathbf{T_c} - \mathbf{T_i})^t}.$$



(a). FACE_CN (b). FACE_CS

(c). FACE_IN (d). FACE_IS

Fig. 1. (a). Original complete normal face, (b). Original complete smiling face, (c). Incomplete normal face, (d). Incomplete smiling face



(a) TEST_A (b) TEST_A: result (c) TEST_B (d) TEST_B: result

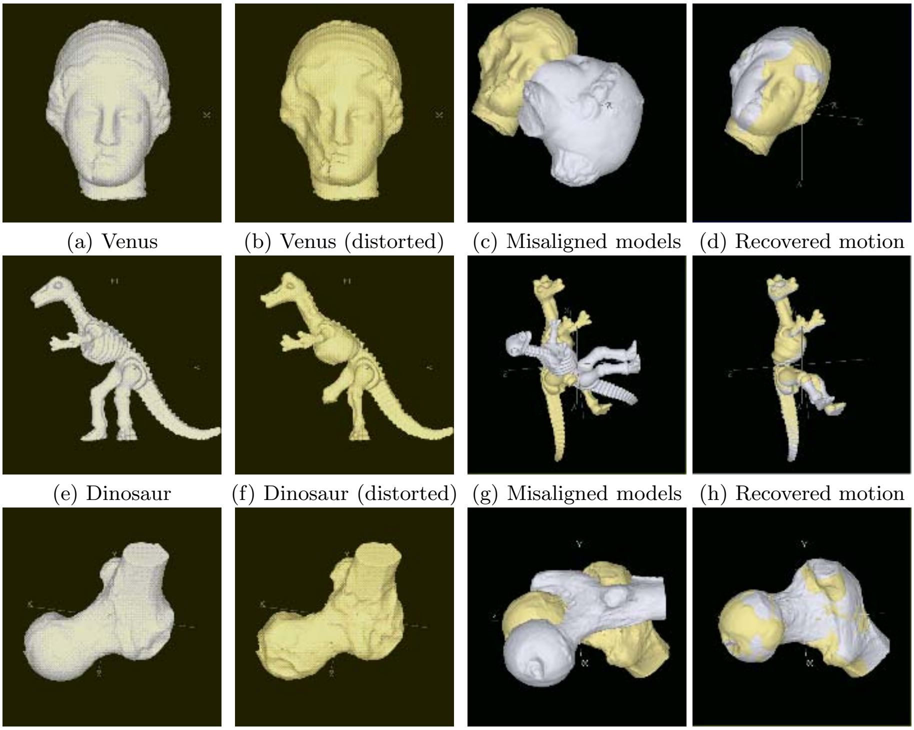
(e) TEST_C (f) TEST_C: result (g) TEST_D (h) TEST_D: result

Fig. 2. Examples of rigid motion estimation results of face data

The rotation error between $\mathbf{R_c}$ and $\mathbf{R_i}$ is computed as

$$\epsilon_r = \sqrt{\mathbf{trace}((\mathbf{R_c} - \mathbf{R_i})(\mathbf{R_c} - \mathbf{R_i})^t)}.$$

Figure 2 visually shows examples of motion recovery results of the face data by our method, and Table 1 shows the average ϵ_t and ϵ_r. Note that in each test group, ϵ_t from all seven transformations are very close to each other, e.g., the variance of seven ϵ_t is less than 10^{-11}. This is true for ϵ_r values also, e.g., the variance of seven ϵ_r is less than 10^{-6}. Figure 3 illustrates examples of results



(a) Venus (b) Venus (distorted) (c) Misaligned models (d) Recovered motion

(e) Dinosaur (f) Dinosaur (distorted) (g) Misaligned models (h) Recovered motion

(i) Ball Joint (j) Ball Joint (distorted) (k) Misaligned models (l) Recovered motion

Fig. 3. Example of rigid motion estimation results of closed surface models

Table 1. Rigid motion error results of face data

Face Tests	$\overline{\epsilon_r}$	$\overline{\epsilon_t}$
TEST_A	10^{-7}	10^{-6}
TEST_B	0.1089	3.8881
TEST_C	0.0008	10^{-5}
TEST_D	0.0928	3.0607

for closed surface data after recovering rigid motions. Table 2 shows the average ϵ_t and ϵ_r of all results of the closed surface data. In each test, ϵ_t from all seven transformations are very close to each other, i.e., the variance of seven ϵ_t is less than 0.6626. This is true for ϵ_r values also, i.e., the variance of seven ϵ_r is less than 0.0042.

From the results of the face data, it can be seen that our method can give accurate rigid motion recovery. In the case where both rigid and nonrigid motions are present, the approach can still give very good estimation of rigid motion. We can also see that incompleteness of the data does not have much influence on the algorithm. The method is robust and gives good estimation of rigid motion

Table 2. Rigid motion error results of closed surface data

Models	$\overline{\epsilon_r}$	$\overline{\epsilon_t}$
Venus	0.0015	0.8911
Isis	0.1155	2.2082
Dinosaur	0.0074	3.1196
Ball Joint	0.0734	1.4513
Hip	0.0008	1.6311
Teeth	0.1418	2.1735

parameters. For the closed surface data, our method gives good estimates for rigid motion, even though matching surfaces are distorted, noisy, and partially missing.

4 Conclusions

We present a hierarchical method for 3D rigid motion estimation which utilizes techniques in multiresolution surface analysis including surface mesh decimation and surface fairing. A 3D mesh segmentation method is used to segment multiresolution surfaces into small patches based on geodesic distance and curvature information. Rigid motions are computed by matching of similar pairs of segments. The similarity of segments is measured by the group of error functions which are invariant to rigid motion. We also use another group of error functions which are variant to rigid motion, to filter and score the transformation results. The best transformation result at the lower resolution surface matching is passed to the higher resolution surface matching for an additional constraint such that the search space in segment selection is reduced. The method was tested to recover rigid motions of various 3D data. From the results, it is shown that our method gives good estimations of rigid motions. The method is also robust in the presence of noise, data distortion, and missing information.

References

1. P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2), 1992.
2. Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3), 1992.
3. J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *Int. J. of Computer Vision and INRIA Technique Report RR 2220*, 18(2), 1996.
4. C. Chua and R. Jarvis. 3d free form surface registration and object recognition. *Int. J. of Computer Vision*, 17(1), 1996.
5. J. Vanden Wyngaerd nad L. Van Gool, R. Koch, and M. Proesmans. Invariant-based registration of surface patches. In *IEEE Int. Conf. on Computer Vision*, 1999.

6. A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(5), 1999.
7. D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. In *Computer Vision and Pattern Recognition*, volume 2, 1999.
8. H. M. Yahia, E. G. Huot, I. L. Herlin, and I. Cohen. Geodesic distance evolution of surfaces: a new method for matching surfaces. In *Computer Vision and Pattern Recognition*, volume 2, 2000.
9. Y. Sun and M. A. Abidi. Surface matching by 3d point's fingerprint. In *IEEE Int. Conf. on Computer Vision*, volume 2, 2001.
10. J. J Koenderink and A. J van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8), 1992.
11. The gnu triangulated surface (gts) library, open source free software. http://gts.sourceforge.net/.
12. P. Lindstrom and G. Turk. Evaluation of memoryless simplification. *IEEE Trans. on Visualization and Computer Graphics*, 5(2), 1999.
13. G. Taubin. Geometric signal processing on polygonal meshes. In *Eurographics'2000*, 2000.

Virtual Fashion Show Using Real-Time Markerless Motion Capture

Ryuzo Okada[1], Björn Stenger[1], Tsukasa Ike[1], and Nobuhiro Kondoh[2]

[1] Corporate Research & Development Center, Toshiba Corporation
[2] Semiconductor Company, Toshiba Corporation
ryuzo.okada@toshiba.co.jp

Abstract. This paper presents a motion capture system using two cameras that is capable of estimating a constrained set of human postures in real time. We first obtain a 3D shape model of a person to be tracked and create a posture dictionary consisting of many posture examples. The posture is estimated by hierarchically matching silhouettes generated by projecting the 3D shape model deformed to have the dictionary poses onto the image plane with the observed silhouette in the current image. Based on this method, we have developed a *virtual fashion show* system that renders a computer graphics-model moving synchronously to a real fashion model, but wearing different clothes.

1 Introduction

In a *virtual fashion show* application the goal is to animate a computer-graphics (CG) model in real-time according to the motion of the real person, while the CG model is wearing a costume different from the actual clothes of the real model. Essentially this task requires an efficient technique for human motion capture with real-time estimation capability.

Currently available commercial motion capture systems require markers or sensors attached to a person. In our system we want to avoid use of visible markers and sensors because fashion models are watched by audiences and we think this is important for variety of motion capture applications in the case of home or office use. One well known approach to vision-based motion capture uses space-carving methods. The shape of a target person is obtained as the intersection of 3D regions generated by inverse projection of silhouettes. This technique [1, 2] requires relatively clean silhouette data obtained from many cameras surrounding the person to be captured. Many approaches that makes use of a 3D shape model of the human body have also been proposed, such as matching feature extracted from captured image and that from the projected 3D shape model [3, 4], learning direct mapping from image features to 3D body pose parameters [5], and defining the force that moves the 3D model to the extracted image feature [6]. These method works with a small number of cameras, but many problems such as stability over long sequences, accuracy, and computational cost remain to be solved. Choosing suitable image feature, such as silhouette [5, 6, 7], depth [4], and edge [3], depending on an individual target application is one of the

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 801–810, 2006.

important issues. Another problem is how to search for the optimal posture in the high-dimensional parameter space. Real-time motion capture has been achieved using incremental tracking, however, in this case the problem of initial posture estimation needs to be solved [8], and often estimation errors can accumulate over long image sequences [9]. The highly nonlinear relationship between similarity and posture parameters further complicates the problem. In order to address this, versions of particle filtering have been suggested [10, 11], which have been shown to yield good results, given manual initialization and off-line processing. Part-based methods [12] or the use of inverse kinematics [13] may be able to solve the initialization problem and reduce the computational cost of the estimation. However, these methods require the localization of individual body parts, which is difficult in cases where self-occlusion occurs and there are few cameras.

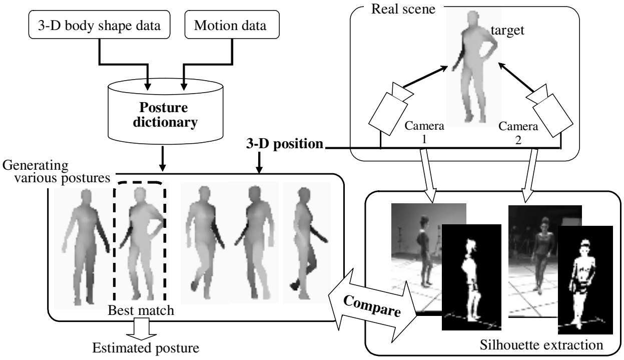


Fig. 1. Overview of the motion capture method

The virtual fashion show application requires real-time processing for synchronizing the motion between the real fashion model and the CG fashion model. Some conditions appropriate for this application can simplify the problem for achieving real-time posture estimation. First, the type of motion is restricted and known beforehand because the motion of the real fashion model is limited to walking and several types of posing. In our setting the fashion model can be required to wear clothes that tightly fit the body, making silhouette matching possible, whose simple definition of cost function also contributes to real-time processing. We are also able to obtain an individual 3D body shape model using a 3D body scanner, as well as posture sequences obtained by a marker-based motion capture system. These data are used to generate a posture dictionary off-line (see section 2 and Fig. 1). Our posture estimation method consists of global position estimation (see section 3) and local pose estimation (see section 4) based on silhouette matching between the observed foreground silhouette and the candidate silhouettes generated from the posture dictionary. We show tracking results and a performance evaluation of posture estimation in section 5 and describe a virtual fashion show in section 6.

2 Posture Dictionary

The 3D body shape model is obtained using a laser 3D shape scanner. The number of polygons is reduced from two million to 2000 by deleting vertexes having small curvatures manually in order to achieve a low computational time for silhouette projection. For 640 × 480 images the time is 1.2–2.0 ms per silhouette projection on a standard PC. The kinematics of the human body are commonly represented by a tree structure whose top node is the body center. Local coordinate systems are defined relative to each body part corresponding to the parent node in the tree structure.

A commercial marker-based motion capture system is used to collect a variety of postures, including walking, posing and turning. A posture captured by the marker-based motion capture system is represented in terms of rotation angles and translation vectors for each joint, which are the parameters to transform a local coordinate system to that of its parent node. Note that the translation parameters are constant except for the body center because the lengths of the body parts do not change, and the parameters of the body center stand for transformation between the local coordinate of the body center and the world coordinate. We call the set of rotation parameters for a posture p a posture vector, which is denoted by $\boldsymbol{r}_p = (r_{p1}, \cdots, r_{p(3N_b)})$, where $N_b = 21$ represents the number of joints.

Due to periodic motion, some poses are very similar, and similar postures are represented by prototype, found by greedy clustering, based on the difference $d_1(a, b)$ between postures a and b:

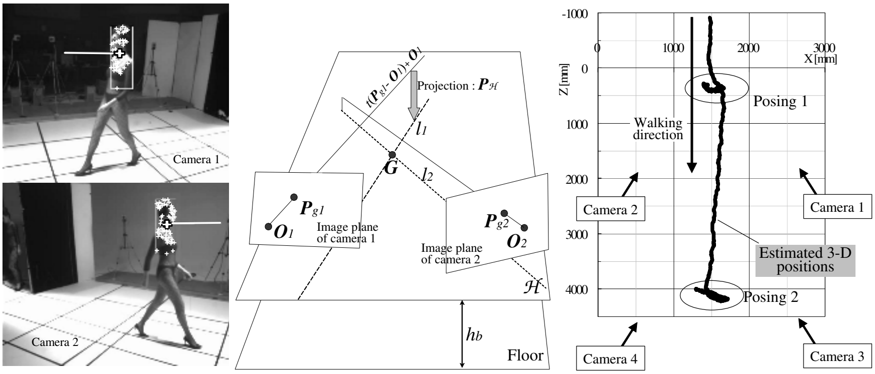
$$d_1(a,b) = \max_{i=1,\cdots,3N_b} |(r_{ai} - r_{bi}) \mod \pi|, \tag{1}$$

which is the largest angle difference of all the rotation parameters. As a result of the clustering, the distances d_1 between any two prototypes are larger than a threshold, which is 7 degrees in our experiments.

3 Global Position Estimation

For estimating the global body position in the 3D scene, we track the target person in two camera views independently based on our previously proposed tracking algorithm [14]. The algorithm enables us to stably track an object in an image sequence captured at a high frame rate as the motion in the image is very small. In our experiments a frame rate of 100 fps is used. The algorithm consists of corner point tracking and outlier elimination using an affine motion model, and estimates the target position in the image as the mean location of the tracked corner points (see Fig. 2(a)).

Next, we compute the global position of the body center in the world coordinate system by triangulation of the two calibrated cameras using the estimated target positions in the images. The postures that we estimate in the virtual fashion show are all upright, so that the body center moves almost parallel to the



(a) Results of feature point tracking (b) Global position estimation (c) Top view of estimated global positions

Fig. 2. Estimation of global positions of a target person. The tracked feature points are indicated by the white '+' marks on the original images in figure (a). The white rectangle is the tracking window, which is the minimum rectangle containing all feature points. The large white '+' marks are the mean positions of the feature points, which is the estimated target position in the image ($\boldsymbol{P}_{g1}$ and $\boldsymbol{P}_{g2}$), and the white line segment attached to it is the estimated motion vector.

floor and the height is approximately fixed to a constant h_b, the height of the body center in standing pose. As shown in Fig. 2(b), we project a line passing through both the camera center $\boldsymbol{O}_c$ and the target position $\boldsymbol{P}_{gc}$ in the image plane onto the plane $\mathcal{H}$, parallel to the floor with distance h_b, and denote the projected line by $\boldsymbol{l}_c$. Assuming that the XZ-plane of the world coordinate system corresponds to the floor, $\boldsymbol{l}_c$ is expressed as follows:

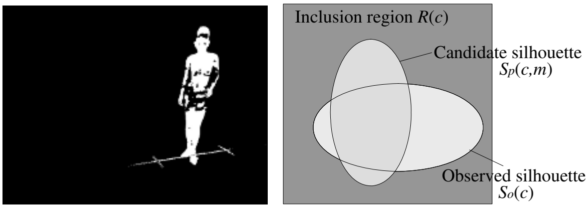
$$\boldsymbol{l}_c = \{\boldsymbol{P}_{\mathcal{H}}(t(\boldsymbol{P}_{gc} - \boldsymbol{O}_{\boldsymbol{c}}) + \boldsymbol{O}_{\boldsymbol{c}}) \mid t \in R\}, \quad c \in \{1,2\}, \ \boldsymbol{P}_{\mathcal{H}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad (2)$$

where $\boldsymbol{P}_{\mathcal{H}}$ denotes the projection matrix onto the plane $\mathcal{H}$. The global position $\boldsymbol{G}$ of the target is the point of intersection of the projected lines $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$.

Fig. 2(c) shows results of global position estimation. The target person walks along the Z-axis at $X = 1500$ and poses at $Z = 500$ and $Z = 4000$ shifting the body weight in the X-direction. In this experiment, two pairs of cameras are used to cover the entire area, but one of them is used for global position estimation at each time instance. The area covered by each pair of cameras is determined beforehand and the pair of cameras is selected when the estimated global position is in its predetermined area.

4 Posture Estimation

We perform the posture estimation procedure at every fourth frame, i.e. at 25 fps, because the computational cost of the posture estimation is much higher than that of global position estimation. First, candidate postures that are in the



(a) Extracted Silhouette (b) Similarity

Fig. 3. Silhouette extraction and similarity computation

neighborhood of the posture in the previous frame are selected as candidates to restrict the search: We select postures that have similar joint angles to the previous posture p, i.e. the distance $d_1(p, m)$ between p and a selected posture m is smaller than a threshold (60 degrees in our experiments). Since the similarity is defined in terms of silhouette difference in the image (see section 4.1 for details), we impose a further restriction on the number of postures based on an appearance-based distance: We define such an appearance-based posture difference, $d_2(a, b)$, using the positions of joints projected onto the image plane for fast computation as

$$d_2(a, b) = \max_{i=1,\cdots,N_b} |\boldsymbol{p}_{ai} - \boldsymbol{p}_{bi}|, \tag{3}$$

where $\boldsymbol{p}_{ai}$ and $\boldsymbol{p}_{bi}$ denote the positions of joints in the image that are obtained by orthogonal projection of the 3D joint coordinates. We sort the postures selected by $d_1(p, m)$ based on the appearance-based distance $d_2(p, m)$, and select the first n postures as the set M of candidate postures. We use $n = 60$ in our experiments. The silhouettes of each candidate posture m in the set M are generated by (1) translating the 3D body shape model to the estimated global position $\boldsymbol{G}$ in order to correspond the size of the silhouette with that of the observed silhouettes, (2) deforming the 3D body shape model to assume the pose m, and (3) projecting the polygons of the deformed 3D body shape model into each camera view.

The observed silhouette is extracted using background subtraction (see Fig. 3(a)). This often results in noisy silhouettes, but has proved sufficiently stable in our application with reasonably stable lighting conditions.

4.1 Similarity of Silhouettes

As shown in Fig. 3(b), $S_p(c, m)$ and $S_o(c)$ denote a candidate silhouette obtained from the candidate posture m, and an observed silhouette for a camera c. $R(c)$ represents the smallest rectangle that contains all candidate silhouettes. The similarity of the silhouettes, $S_p(c, m)$ and $S_o(c)$, should be high when the area of the observed silhouette is large in the candidate silhouette and is small outside the candidate silhouette. Thus, we use the difference between the occupancy rate of the observed silhouette in the candidate silhouette $\rho_i(c, m)$ and that outside the candidate silhouette $\rho_o(c, m)$ for the similarity normalized with the area of the silhouette:

$$\rho_i(c,m) = \frac{|S_p(c,m) \cap S_o(c)|}{|S_p(c,m)|}, \quad \rho_o(c,m) = \frac{|\overline{S_p(c,m)} \cap R(c) \cap S_o(c)|}{|\overline{S_p(c,m)} \cap R(c)|}, \tag{4}$$

where $|\cdot|$ represents the area of a region.

The similarity measure is affected by the estimation error of the global position. It is therefore necessary to perform optimization for both posture and local shift of the global position. We shift the candidate silhouette in each camera view with a shift $\boldsymbol{d}$, and maximize the similarities independently for each camera in order to optimize the global position locally. Thus, we redefine the similarity for a posture m as

$$s(m) = \sum_c \max_{\boldsymbol{d} \in D} \; (\rho_i(c,m,\boldsymbol{d}) - \rho_o(c,m,\boldsymbol{d})), \tag{5}$$

where $\rho_i(c,m,\boldsymbol{d})$ and $\rho_o(c,m,\boldsymbol{d})$ denote the occupancy rate using a candidate silhouette shifted with a shift $\boldsymbol{d}$ in the range of shifts D.

4.2 Hierarchical Posture Search

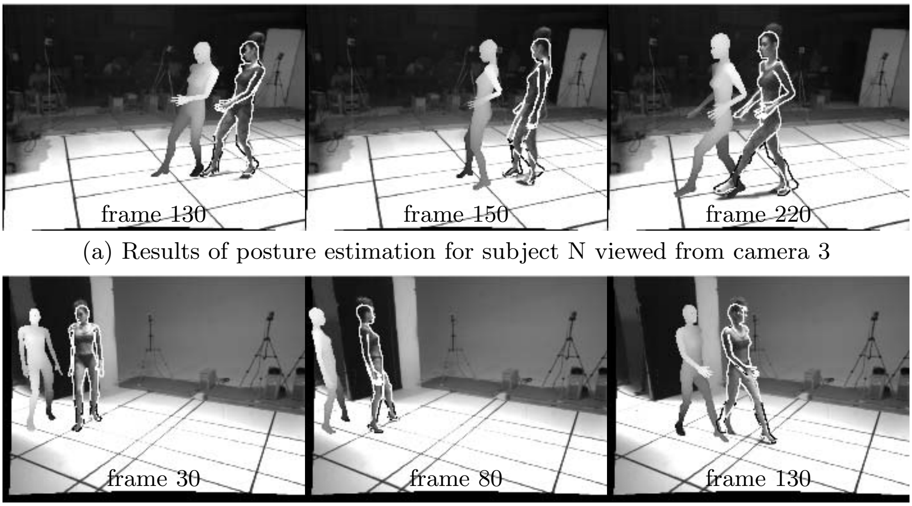
In order to reduce the computational cost of searching for the posture with the greatest similarity, we adopt a coarse-to-fine strategy using a two-level tree, which is generated on-line for each frame. The first layer of the search tree consists of postures selected from M at every t-th posture and the rest of the candidate postures are attached to the closest posture in the first layer as postures in the second level. We search for the optimal posture using the search tree as follows: (1) compute the similarity based on eq. (5) for the postures on the first level of the tree, (2) select the k postures with the greatest similarity, (3) compute the similarity for the postures on the second level in the subtrees of the k selected postures, and (4) select the posture that has the greatest similarity. We use $t = 3$ and $k = 3$ in our experiments.

4.3 Initialization

If a sufficiently large silhouette is extracted in the current image based on the background subtraction, we set the observed silhouette to be the initial target region and start tracking based on our object tracking algorithm [14].When the tracking results come from two or more cameras for the first time, we compute the initial global position, and start the posture estimation with suitable initial posture. Although the initial posture does not fit completely to the posture of the target person, the estimated posture gradually fits to the target person in the subsequent frames.

5 Experiments

Fig. 4 shows the results of posture estimation using four cameras for two subjects who walk and pose differently. In each case the camera arrangement is the same



(a) Results of posture estimation for subject N viewed from camera 3

(b) Results of posture estimation for subject S viewed from camera 2

Fig. 4. Results of posture estimation using four cameras. Estimated postures are indicated by white contour lines on the original image. The 3D figures shown in gray beside the contours represent postures with smoothed motion for CG animation.

as that shown in Fig. 2(c) and the cameras capture gray-scale images with a resolution of 640x480 pixels at a frame rate of 100 fps. Note that the frame rate for posture estimation is 25 fps as described in section 4. The posture sequence for each subject obtained by the marker-based motion capture system contains about 3600 postures captured at a rate of 120 fps. After clustering the posture vectors the dictionary for each subject consists of about 500 postures, which are experimentally sufficient for our restricted set of motions in the virtual fashion show. Fig. 4 shows that postures are correctly estimated in most frames. In some frames, such as frame 150 in Fig. 4(a), however, the contour lines showing the estimated postures are incorrect.

We have conducted experiments on 27 image sequences for evaluating the performance of the posture estimation. The image sequences include four types of motion shown in Fig. 5(b) performed by three subjects. We use an individual 3D body shape model and motion data for each subject obtained by a laser 3D shape scanner and a commercial marker-based motion capture system, respectively. Table 1 shows the number of misestimations and the number of frames in which misestimation occurs. The number of misestimations, e.g. frame 150 in Fig. 4(a), is counted by comparing the estimate to the ground truth, where postures with small alignment errors are not counted as a misestimation. The misestimation often occurs for a particular subject H compared to the other subjects. This is because her postures in the image sequences are not contained in the posture dictionary. In total misestimation occurs 34 times for 27 sequences, on average about 1.3 times per sequence (0.089 times per second), corresponding to 4.3 % of the total number of frames. Although we have restricted the

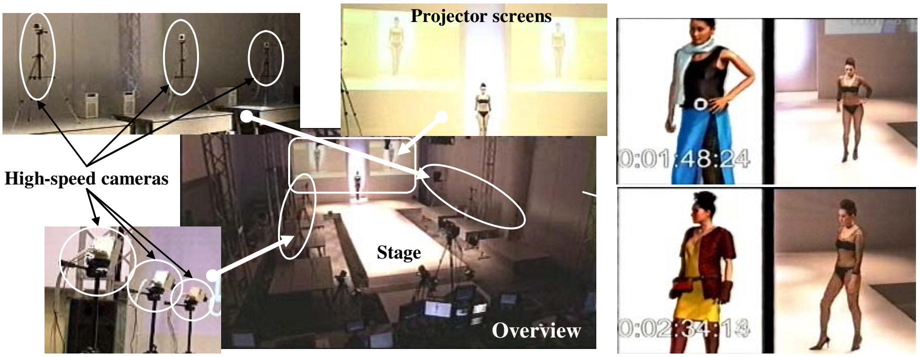
Table 1. Performance evaluation. The first column represents the type of scenario. For example, S-M1 stands for motion sequence M1 performed by subject S. The second column is the number of sequences, which are used in the experiments, and the third column is the total number of frames. Columns four to six show the number of misestimations, the number of frames in which misestimation occurs and the error rate.

Scenario	# Sequence	Frames	Failures	Failure frames	Error in %
S-M1	4	1318	4	12	0.9
S-M2	3	1120	4	87	7.8
S-M3	3	1017	1	17	1.7
S-M4	4	1414	5	37	2.6
H-M2	4	1593	7	127	8.0
H-M3	4	1475	6	64	4.3
N-M1	3	924	6	62	6.7
N-M4	2	694	1	5	0.7

search space for posture estimation by selecting candidate postures similar to the previous estimated posture, misestimation occurs for a short period. Such temporal jitter can be reduced by temporal filtering. In our system smooth motion is generated based on the posture sequence recorded by a marker-based motion capture system (see section 6.1). Another reason for the misestimations is the fact that the extracted silhouettes can be very noisy due to shadows on the floor.

6 Virtual Fashion Show

We have developed a virtual fashion show system using our motion estimation method described in sections 2–4. Fig. 5 shows an overview of the system. A fashion model walks and poses on the stage according to four types of scenarios



(a) Virtual fashion show set up (b) Examples of scenarios

Fig. 5. Overview of the virtual fashion show. Three pairs of cameras are placed along both sides of the stage. Two projectors show the CG model wearing clothes different from the actual model as shown in the left images of figure (b).

shown in Fig. 5(b), and our motion capture system estimates her posture. While the fashion model walks along the stage, she poses twice at different positions according to the scenario. Two large projector screens display a full-CG model wearing a costume different from the actual clothes, based on clothes simulation and CG techniques.

6.1 Smooth Motion Generation

As described in section 5, misestimation of the posture occurs at a certain rate. Even when the posture is correctly estimated, the estimated motion, which is the time series of the estimated postures, is not smooth because the estimated postures can be slightly misaligned. These problems are critical for generating natural motion of the CG model in the virtual fashion show. We thus combine the estimated motion with the motion data recorded with the marker-based motion capture system.

The recorded motion sequence contains all the postures in the same order as the motion of a real fashion model, except for the timing of walking and posing. We generate smooth motion by changing the playing speed of the recorded posture sequence according to the estimated posture. We start playing the recorded posture sequence when the current estimated posture e is similar to the posture i in the first frame of the recorded posture sequence in terms of the posture difference $d_1(e, i)$.

The 3D figures shown in gray in Fig. 4 represent postures generated by the smooth motion generation method. In the 150th frame in Fig. 4(a) where the posture is misestimated, the motion model finds a plausible posture, even if the silhouette of the estimated posture is slightly misaligned with the observed silhouette. While this smooth motion generation method is straightforward and effective for a specific application of the virtual fashion show, accurate posture estimation and a universal motion generation method are necessary for general applications.

6.2 Hardware Configuration

We place three high-speed cameras on each side of the stage (six cameras in total) in order to cover the entire stage which measures about 10 m × 3 m. Each high-speed camera is connected to a PC mounting dual Xeon 3.0 GHz CPUs that captures images and tracks the fashion model in the images as described in section 3. The captured images and the tracking results are transfered to two PCs for posture estimation mounting quad Itanium 2 1.6 GHz CPUs through a high-speed network *Myrinet*, and the two PCs compute the global position and estimate the posture with smooth motion generation. The estimated posture is sent to a PC for clothes simulation and CG rendering through Gb Ethernet, and the generated CG animation is displayed on two projector screens.

7 Conclusions and Future Work

We have presented a real-time motion capture system using pairs of cameras and have demonstrated that the system works efficiently for a virtual fashion show based on several constraints appropriate for the virtual fashion show, such as known body shape, tight fitting clothes and limited types of motion.

A possible future application is a virtual try-on for online clothes shopping. However, in order to make this approach work in more general settings, some issues that need to be considered are automatic 3D body shape model acquisition, the use of more robust image features, and efficient matching techniques for increasing the number of postures in the posture dictionary.

References

1. Matsuyama, T., Wu, X., Takai, T., Wada, T.: Real-time dynamic 3D object shape reconstruction and high-fidelity texture mapping for 3D video. IEEE Trans. on Circuits and Systems for Video Technology **14** (2004) 357–369
2. Cheung, G., Baker, S., Kanade, T.: Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In: Proc. of CVPR. Volume 1. (2003) 77–84
3. Gavrila, D., Davis, L.: 3d model-based tracking of humans in action: A multi-view approach. In: Proc. of CVPR. (1996) 73–80
4. Plänkers, R., Fua, P.: Tracking and modeling people in video sequences. Computer Vision and Image Understanding **81** (2001)
5. Agarwa, A., Triggs, B.: 3d human pose from silhouettes by relevance vector regression. In: Proc. of CVPR. Volume 2. (2004) 882–888
6. Delamarre, Q., Faugeras, O.: 3d articulated models and multi-view tracking with silhouettes. In: Proc. of ICCV. Volume 2. (1999) 716–721
7. Brand, M.: Shadow puppetry. In: Proc. of ICCV. (1999) 1237–1244
8. Senior, A.: Real-time articulated human body tracking using silhouette information. In: Proc. of IEEE Workshop on Visual Surveillance/PETS. (2003) 30–37
9. Yamamoto, M., Ohta, Y., Yamagiwa, T., Yagishita, K., Yamanaka, H., Ohkubo, N.: Human action tracking guided by key-frames. In: Proc. of FG. (2000) 354–361
10. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: Proc. of CVPR. Volume 2. (2000) 1144–1149
11. Sminchisescu, C., Triggs, B.: Estimating articulated human motion with covariance scaled sampling. IJRR **22** (2003) 371–391
12. Felzenszwalb, P., Huttenlocher, D.: Efficient matching of pictorial structures. In: Proc. of CVPR. Volume 2. (2000) 66–73
13. Date, N., et al.: Real-time human motion sensing based on vision-based inverse kinematics for interactive applications. In: Proc. of ICPR. Volume 3. (2004) 318–321
14. Okada, R., et al.: High-speed object tracking in ordinary surroundings based on temporally evaluated optical flow. In: Proc. of IROS. (2003) 242–247

Space-Time Invariants for 3D Motions from Projective Cameras

Ying Piao and Jun Sato

Department of Computer Science and Engineering, Nagoya Institute of Technology,
Nagoya 466-8555, Japan
paku@hilbert.elcom.nitech.ac.jp, junsato@nitech.ac.jp

Abstract. Recently, it has been shown that invariants on motions can be extracted from sequential images and these can be applied for recognizing dynamic events from images viewed from arbitrary viewpoints. These invariants are called space-time invariants since they are defined in space and time. Unfortunately, the existing space-time invariants are limited for planar motions viewed from affine cameras. In this paper, we propose a method for computing space-time invariants on general 3D motions viewed from projective cameras. Furthermore, we show that by using the epipolar geometry derived from the mutual projection of cameras, the stability of space-time invariants can be improved drastically. The extracted invariants are applied for distinguishing non-rigid 3D motions from video sequences viewed from arbitrary viewpoints.

1 Introduction

For recognizing dynamic events, it is important to analyze visual events both in space and time domains [4, 2, 1]. However, it is difficult to recognize motions from conventional statistical methods if the viewpoints of cameras are arbitrary. Even if we can sometimes use assumptions on the structure of moving objects, such as articulated motions[9, 6], these assumptions limit the class of objects or events to be recognized.

For recognizing objects from arbitrary viewpoints, geometric invariants are very useful [7, 10, 13, 11], and Levin et al. showed the invariance on multiple points with constant motions [5]. For recognizing general motions, Sato proposed the space-time invariants [8]. The important properties of the space-time invariants are that they are identical even if the image motions are viewed from arbitrary viewpoints, and thus they can be applied for recognizing dynamic events from arbitrary views. However, the existing space-time invariants are limited in two folds. Firstly, the dynamic actions, which can be recognized from the existing space-time invariants are limited to planar motions. Secondly, the existing space-time invariants assume affine projections or weak perspective projections, and thus they cannot be applied if we have strong perspective distortions in images.

In this paper, we propose a method for computing space-time invariants for general non-rigid 3D motions from image sequences viewed from projective cameras. In particular, we show that space-time invariants for non-rigid 3D motions

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 811–821, 2006.

can be computed from a set of 6 points in sequential images observed from arbitrary viewpoints. We also show that the stability of space-time invariants can be improved drastically by using the epipolar geometry derived from the mutual projection of cameras. The extracted invariants are applied for distinguishing non-rigid 3D motions from video sequences viewed from arbitrary viewpoints.

2 Invariants on 3D Motions

The motions of a point, $\mathbf{X} = [X, Y, Z]^\top$, in a 3 dimensional space, $\mathit{\Pi}^3$, can be considered as a set of points, $\mathbf{W} = [X, Y, Z, t]^\top$, in 4 dimensional space-time, $\mathit{\Pi}^3 \times \mathit{\Sigma}$. The motions in the real space are projected to images, and can be observed as a set of points, $\mathbf{w} = [x, y, t]^\top$, in a 3 dimensional space-time $\pi^2 \times \sigma$ on image motions.

The 4 dimensional information, $\mathbf{W} = [X, Y, Z, t]^\top$, is required for computing space-time invariants of general 3D motions. However, we have only 3 dimensional information $\mathbf{W} = [X, Y, t]^\top$ from a single camera image. Thus, a single camera is not enough for computing projective space-time invariants of general 3D motions.

Since 3 dimensional information $\mathbf{w}$ is available from a single camera, the necessary condition for computing invariants in 4D space-time from N cameras is $3N \geq 4$. Thus, two or more than two cameras are required for computing the 4D projective space-time invariants. In this paper, we consider a method for computing the 4D projective space-time invariants from two projective cameras.

3 Projective Bases and Projective Depth

Since the computation of projective depth is important for computing 4D space-time invariants, we quickly review a method for computing projective depth from the epipolar geometry.

Let a point, $\mathbf{X} = [X, Y, Z]^\top$, in the 3D space be projected to image points, $\mathbf{x} = [x, y]^\top$ and $\mathbf{x}' = [x', y']^\top$, by two projective cameras at two different viewpoints as follows:

$$\lambda\widetilde{\mathbf{x}} = \mathbf{P}\widetilde{\mathbf{X}} \qquad \lambda'\widetilde{\mathbf{x}}' = \mathbf{P}'\widetilde{\mathbf{X}} \tag{1}$$

where, $(\tilde{\ })$ denotes homogeneous coordinates, and $\mathbf{P}$ and $\mathbf{P}'$ denote 3×4 projection matrices.

The epipolar geometry between these two cameras can be computed nonlinearly from 7 corresponding points [12] and linearly [3] from 8 corresponding points in images. Once the epipolar geometry is obtained, we can calibrate these two cameras with respect to some specific projective frames, and we can reconstruct 3D points, $\mathbf{X}$, up to a projective ambiguity. Then the projective depth, λ and λ', can be obtained with respect to the projective frame. The projective frames can be defined by a set of 5 basis points. Although we can choose any 5 points as bases, it is important to choose two viewpoints as two of 5 basis points for defining space-time invariants later. The remaining 3 basis points can be chosen from general 3D points freely, but it is better to choose these points from the

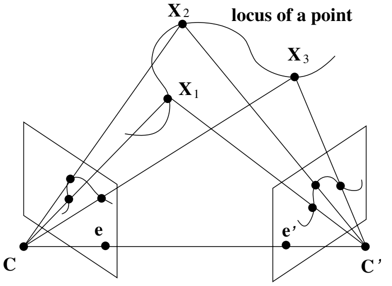


Fig. 1. The projective bases for computing the projective depth of each point on 3D motions. Two viewpoints and three points on the 3D motions are chosen as the projective bases.

3D motions (i.e. loci of points), since in this case we do not need to consider additional points in the scene. Thus, we choose 5 basis points, $\{\mathbf{C}, \mathbf{C}', \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\}$ as shown in Fig. 1, and give them standard basis coordinates, such as $[0, 0, 0, 1]^\top$, $[1, 1, 1, 1]^\top$, $[1, 0, 0, 0]^\top$, $[0, 1, 0, 0]^\top$ and $[0, 0, 1, 0]^\top$. Then, we can derive projective depth, λ and λ', with respect to these projective bases.

4 Projective Space-Time Invariants for 3D Motions

We next consider a method for computing projective space-time invariants for 3D motions.

Suppose a point $\mathbf{W} = [X, Y, Z, t]^\top$ in the real space-time $\Pi^3 \times \Sigma$ is projected to a point $\mathbf{w} = [x, y, t]^\top$ in the image space-time $\pi^2 \times \sigma$ by a projective camera. Then, this space-time projection can be described as follows:

$$\begin{bmatrix} \lambda x \\ \lambda y \\ t \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & p_{14} \\ p_{21} & p_{22} & p_{23} & 0 & p_{24} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ t \\ 1 \end{bmatrix} \tag{2}$$

Although the space-time projection from $\mathbf{W}$ to $\mathbf{w}$ cannot be described by affine cameras nor projective cameras, (2) describes that a point $\mathbf{W}$ in the real space-time is projected to a point $[\lambda x, \lambda y, t]^\top$ by an extended affine camera.

If the point, $\mathbf{W}$, is also projected to $\mathbf{w}' = [x', y', t]^\top$ in another viewpoint, the space-time projections of these two cameras can be described as follows:

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda' x' \\ \lambda' y' \\ t \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & p_{14} \\ p_{21} & p_{22} & p_{23} & 0 & p_{24} \\ p'_{11} & p'_{12} & p'_{13} & 0 & p'_{14} \\ p'_{21} & p'_{22} & p'_{23} & 0 & p'_{24} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ t \\ 1 \end{bmatrix} \tag{3}$$

From (3), we find that the relationship between $[\mathbf{W}^\top, 1]^\top$ and $[\lambda x, \lambda y, \lambda' x', \lambda' y', t]^\top$ can be described by affine transformations in 5D space with no translations. Thus, affine invariants computed from $[\mathbf{W}^\top, 1]^\top$ coincide with affine invariants computed from $[\lambda x, \lambda y, \lambda' x', \lambda' y', t]^\top$. Therefore, if we know the projective depth, λ and λ', from the projective reconstruction of a point $\mathbf{X}$, then the projective space-time invariants for 3D motions, $\mathcal{I}$, can be defined as affine invariants in 5D space as follows:

$$\mathcal{I} = \frac{\begin{vmatrix} \lambda_i \mathbf{x}_i & \lambda_j \mathbf{x}_j & \lambda_k \mathbf{x}_k & \lambda_l \mathbf{x}_l & \lambda_n \mathbf{x}_n \\ \lambda'_i \mathbf{x}'_i & \lambda'_j \mathbf{x}'_j & \lambda'_k \mathbf{x}'_k & \lambda'_l \mathbf{x}'_l & \lambda'_n \mathbf{x}'_n \\ t_i & t_j & t_k & t_l & t_n \end{vmatrix}}{\begin{vmatrix} \lambda_i \mathbf{x}_i & \lambda_j \mathbf{x}_j & \lambda_k \mathbf{x}_k & \lambda_l \mathbf{x}_l & \lambda_m \mathbf{x}_m \\ \lambda'_i \mathbf{x}'_i & \lambda'_j \mathbf{x}'_j & \lambda'_k \mathbf{x}'_k & \lambda'_l \mathbf{x}'_l & \lambda'_m \mathbf{x}'_m \\ t_i & t_j & t_k & t_l & t_m \end{vmatrix}} \tag{4}$$

Where, $|\cdot|$ denotes the determinant of a 5×5 matrix which consists of five vectors, $[\lambda \mathbf{x}^\top, \lambda' \mathbf{x}'^\top, t]^\top$. The index i, j, k, l, m, and n of these vectors can be chosen from the permutation of $1, \cdots, 6$, but the number of functionally independent space-time invariants $\mathcal{I}$ is only 5.

Note, in general we need 7 points for defining affine invariants in 5D space, but we need only 6 points in this case, since there is no translation component in the affine transformation in (3).

However, we have to note one important thing. That is the ambiguity of the projective depth, λ and λ'. In general, the projective depth changes projectively, if we choose different sets of basis points. Thus, the affine invariants defined by (4) is no longer invariant if we change projective bases. However, if we choose two viewpoints as two of five basis points as shown in section 3, the changes in projective depth caused by the changes in viewpoints can be described by affine transformations. Thus, the space-time invariant defined by (4) is invariant under the changes in viewpoints.

The proposed space-time invariants are different from simple invariants on loci without time. Suppose we have two motions which have the same loci in the 3D space but have different speed patterns, for example one has a constant speed and another has non-constant speed. Although these two motions have the same loci, we consider these two are different motions, since the speed patterns are different. If we simply consider invariants on loci without considering the time domain explicitly, these two motions have the same invariants and we cannot distinguish them. However, if we compute space-time invariants proposed in this paper, the invariants computed from these two motions have different values and we can distinguish these two motions. Thus the proposed space-time invariants are very useful for distinguishing motions from arbitrary viewpoints.

5 Invariants on Non-rigid Motions of Multiple Points

Up to now we find that the space-time invariants can be computed from 6 frame motions of a single point. If we have more moving points in the space, the space-time invariants can be computed from less image frames. The important point here is that the motions of these multiple points can be non-rigid. That is, by using the proposed method, non-rigid 3D motions of multiple points can be recognized from arbitrary views.

Since we can compute space-time invariants from any set of 6 points in the 4D space-time, the space-time invariants can be derived if the following condition holds:

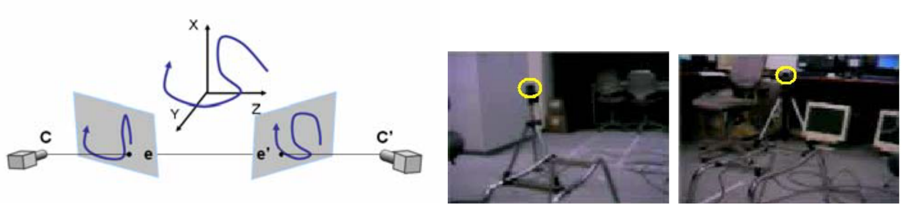
$$N_p \times N_f \geq 6 \tag{5}$$

where, N_p denotes the number of points in the 3D space, and N_f denotes the number of frames. This means we can also compute projective space-time invariants from two points with three frames or three points with two frames. These invariants enable us to distinguish non-rigid 3D motions of multiple motions from arbitrary views.

6 Epipolar Geometry from Mutual Projection of Camera

Up to now, we have seen that we can compute space-time invariants on 3D motions from image sequences viewed from arbitrary multiple viewpoints. In this section, we consider mutual projection of cameras in images and show that we can further stabilize the projective space-time invariants by using the mutual projection of multiple cameras.

We consider the case, where two cameras for computing invariants are projected each other as shown in Fig. 2 (a). If the projection of a camera is small enough, the center of the projected camera in images can be considered as the projection of a viewpoint, $\mathbf{C}'$, to the other viewpoint, $\mathbf{C}$. In this case, we can directly obtain an epipole, $\mathbf{e} = [e_u, e_v, e_w]^\top$, as well as $\mathbf{e}' = [e'_u, e'_v, e'_w]^\top$, from the projection of cameras $\mathbf{C}$ and $\mathbf{C}'$. As shown in Fig. 2, the projected cameras are in general enough small to be considered as a viewpoint, and this approximation is valid in most of the case.



(a) mutual projection of two cameras (b) $\mathbf{C}'$ in image 1 (c) $\mathbf{C}$ in image 2

Fig. 2. Epipolar geometry from mutual projection of cameras

Since the relationships between the epipoles, $\mathbf{e}$ and $\mathbf{e}'$, and the fundamental matrix, $\mathbf{F}$, can be described by $\mathbf{F}\widetilde{\mathbf{e}} = \mathbf{0}$ and $\mathbf{F}^\top\widetilde{\mathbf{e}} = \mathbf{0}$, we have the following equation on the components of $\mathbf{F}$:

$$\mathbf{M}_e\mathbf{f} = \mathbf{0} \tag{6}$$

where, $\mathbf{f}$ is a 9 vector which consists of the 9 components of $\mathbf{F}$, and $\mathbf{M}_e$ is a 6×9 matrix as follows:

$$\mathbf{M}_e = \begin{bmatrix} e_u & e_v & e_w & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e_u & e_v & e_w & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e_u & e_v & e_w \\ e'_u & 0 & 0 & e'_v & 0 & 0 & e'_w & 0 & 0 \\ 0 & e'_u & 0 & 0 & e'_v & 0 & 0 & e'_w & 0 \\ 0 & 0 & e'_u & 0 & 0 & e'_v & 0 & 0 & e'_w \end{bmatrix}$$

Although the matrix $\mathbf{M}_e$ is 6×9, its rank is five. If we have 3 corresponding points, $\mathbf{m}_i$ and $\mathbf{m}'_i$ $(i = 1, \cdots, 3)$, in images besides the given two epipoles, we have 3 more linear equations, $\mathbf{M}_p\mathbf{f} = \mathbf{0}$, from $\mathbf{m}'^{\top}_i\mathbf{F}\mathbf{m}_i = 0$. and $\mathbf{f}$ can be computed by solving the following linear equations:

$$[\mathbf{M}_e^{\top}, \mathbf{M}_p^{\top}]^{\top}\mathbf{f} = \mathbf{0} \tag{7}$$

where, $\mathbf{M}_p$ is a 3×9 matrix which consists of the components of $\mathbf{m}$ and $\mathbf{m}'$. If we have more than 3 corresponding points, we compute least square solutions to $[\mathbf{M}_e^{\top}, \mathbf{M}_p^{\top}]^{\top}\mathbf{f} = \mathbf{0}$ subject to $\mathbf{M}_e\mathbf{f} = \mathbf{0}$.

Although the 8 points algorithm provides us good epipolar geometry from general point constellations, it is inaccurate and unstable if the corresponding points are close to coplanar in the 3D space. This means if the 3D motions are close to planar motions, the space-time invariants computed by using the 8 point algorithm are sensitive to image noises. On the other hand, by using the mutual projection of cameras, we can compute accurate and stable epipolar geometry, even if the 3D points are coplanar. Thus we can extract accurate space-time invariants even if the 3D motions are close to planar motions.

7 Experiments

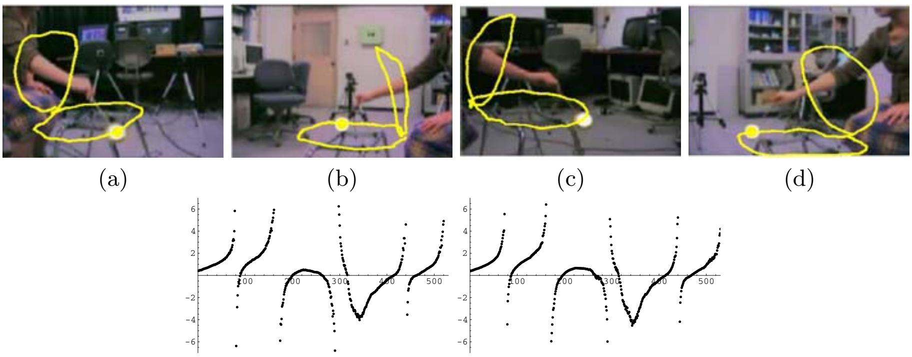
We next show the results from some real image experiments and efficiency analysis of the proposed invariants on 3D motions.

7.1 Real Image Experiments

We first show the results from some real image experiments. Fig. 3 (a) and (b) show a pair of image motions extracted from a single 3D motion, and (c) and (d) show another pair of image motions extracted from the same 3D motion. These image motions are extracted by using a correlation tracker. As shown in these figures, the image motions are very different if the viewpoints are different. The projective space-time invariants are computed from Fig. 3 {(a), (b)} and Fig. 3 {(c), (d)} respectively. Fig. 3 (e) shows a projective space-time invariant signature computed from (a) and (b), and Fig. 3 (f) shows that from (c) and (d). As shown in these figures, the projective space-time invariants are almost identical, even if the image motions are very different as shown in Fig. 3 (a), (b), (c) and (d). As described in section 4, we have 5 functionally independent space-time invariants and we used invariants whose index is $\{i, j, k, l, m, n\} = \{1, 2, 3, 4, 5, 6\}$ in our experiments.

Fig. 4 (a), (b), (c) and (d) show image motions extracted from another 3D motion, and (e) and (f) show projective space-time invariants computed from {(a), (b)} and {(c), (d)} respectively. As shown in Fig. 4 (e) and (f), the projective space-time invariants are almost identical again.

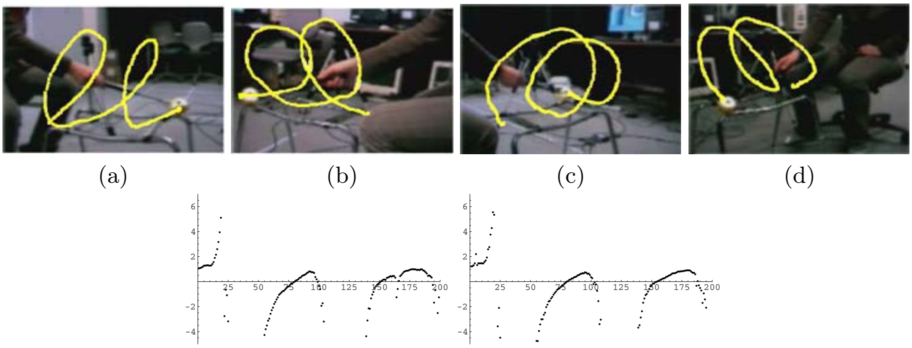
As shown in Fig. 3 and Fig. 4, the projective space-time invariants are very different if the motions are different each other. From these figures we find that the proposed space-time invariants are very useful for distinguishing different motions from video sequences extracted from arbitrary viewpoints.



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

(e) (f)

Fig. 3. The image motions at four different viewpoints. (a), (b), (c) and (d) show image motions at four different viewpoints. (e) shows projective space-time invariant signatures computed from (a) and (b). (f) shows those from (c) and (d).



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

(e) (f)

Fig. 4. The image motions at four different viewpoints and projective space-time invariants. (a), (b), (c) and (d) show image motions at four different viewpoints. (e) shows a projective space-time invariant signature computed from (a) and (b), and (f) shows that computed from (c) and (d).

Table 1. Coincidence degrees e computed from two space-time invariant signatures

two invariant signatures	e
Fig.3 (e) and (f)	0.0999
Fig.4 (e) and (f)	0.0613
Fig.3 (e) and Fig.4 (e)	1.6838

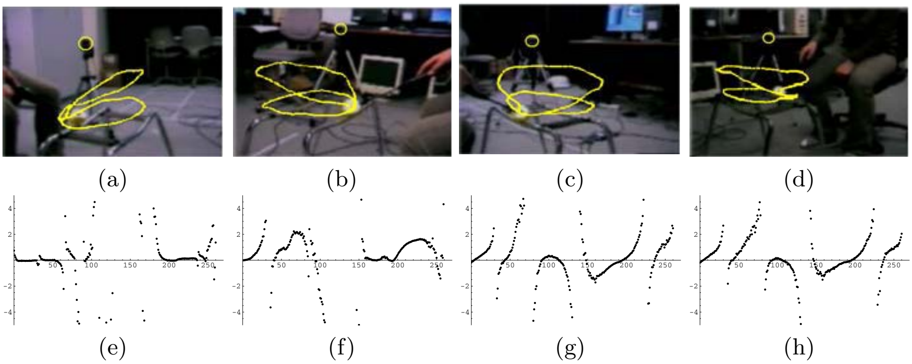
We next evaluate the similarity of invariant signatures numerically. We define a coincidence degree of two invariant signatures, $\mathcal{I}$ and $\mathcal{I}'$, as follows:

$$e = \frac{1}{N}\sum_{t=1}^{N}(\mathcal{I}(t) - \mathcal{I}(t)')^2 \tag{8}$$

where, N denotes the total number of frames in invariant signatures.

Table 1 shows coincidence degrees computed from two invariant signatures of motion 1, two invariant signatures of motion 2, and invariant signatures of motion 1 and 2 respectively. As shown in this table, the coincidence degrees computed from the same motions are very small, while the coincidence degree computed from different motions is large.

Up to now, we have computed the space-time invariants on 3D motions from the 8 points algorithm. We next compare the stability of projective space-time invariants computed from the 8 points algorithm and the mutual projection algorithm. In this experiment we used 3D motions which are close to planar motions. Fig. 5 (a), (b), (c) and (d) show image motions at four different view points. Fig. 5 (e) and (f) show space-time invariants computed by using the 8 points algorithm, and (g) and (h) show those from the mutual projection algorithm. As shown in (e) and (f), the invariants computed from the 8 points algorithm is far from identical, although they are computed from the same motion. This is because the epipolar geometry computed from nearly coplanar 3D



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

(e) (f) (g) (h)

Fig. 5. The space-time invariants computed from a 3D motion which is close to coplanar. (a), (b), (c) and (d) show image motions at four different viewpoints. (e) and (f) show projective space-time invariants computed from ((a), (b)) and ((c), (d)) respectively by using the 8 points algorithm. (g) and (h) show those computed by using the mutual projection of cameras.

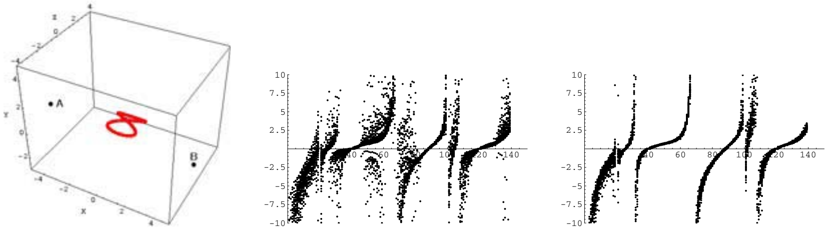
points is in general very unstable. On the other hand, the space-time invariants computed from the mutual projection algorithm is almost identical, as shown in (g) and (h). From these results, it is clear that the stability of projective space-time invariants is drastically improved by using the mutual projection algorithm. Thus, if we have the projection of cameras in images, we had better use the information for computing invariants.

7.2 Stability Evaluation

We next compare the stability of space-time invariants computed from the 8 points algorithm and the mutual projection algorithm by using a synthetic 3D motion which is close to coplanar. The red curve in Fig. 6 (a) shows a synthetic 3D motion used in this experiment. (A) and (B) show viewpoints of the pair of cameras. The invariant signatures are extracted from images viewed from these two cameras adding Gaussian image noises with the standard deviation of 1 pixel. Fig. 6 (b) and (c) show space-time invariants computed from the 8 points algorithm and the mutual projection algorithm respectively. As shown in these figures, the uncertainty bound of the space-time invariants computed from the mutual projection algorithm is very small comparing with those from the 8 points algorithm. This agrees the results in Fig. 5.

7.3 Non-rigid 3D Motions

Finally, we show the results from non-rigid 3D motions of multiple points. Fig. 7 (a) and (b) show two different motions of Japanese “Karate”. The 4 markers are put on two hands and two legs, and image motions of these 4 markers are taken by using a correlation tracker. The color lines in (a) and (b) show extracted image motions of these 4 points. Fig. 7 (c) and (d) show the space-time representation of motion 1 viewed from two different viewpoints, and (e) and (f) show those



(a) (b) (c)

Fig. 6. (a) shows an example 3D motion which is close to coplanar. The red curve shows a synthetic 3D motion. (A) and (B) show viewpoints of two cameras. (b) shows the uncertainty bound of projective space-time invariants computed from a pair of cameras shown in (a) by using the 8 points algorithm. (c) shows the uncertainty bound of projective space-time invariants extracted by using the mutual projection of cameras. These invariants are computed by adding Gaussian noises with the standard deviation of 1 pixel.

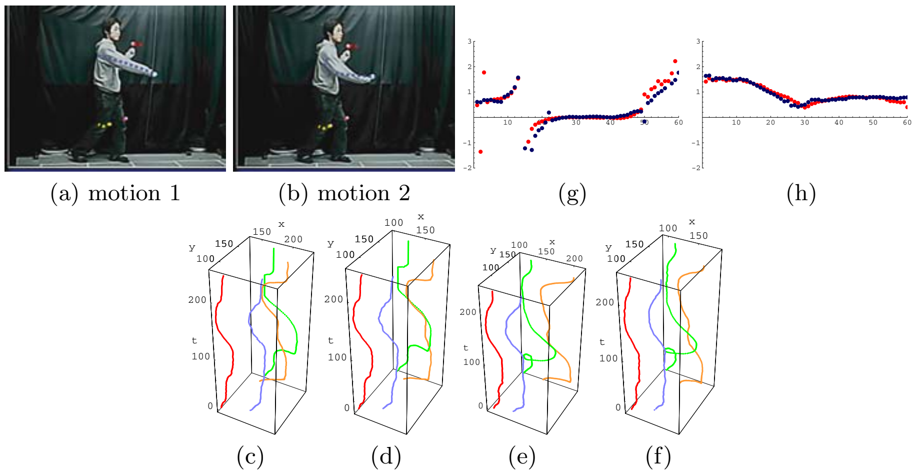


Fig. 7. The space-time invariants computed from non-rigid 3D motions of multiple points. (a) and (b) show two different motions of "Karate". The four markers are put on two hands and two legs and are tracked in images. (c) and (d) show the space-time representation of motion 1 viewed from two different viewpoints. (e) and (f) show those from motion 2. The red and blue curves in (g) show space-time invariants on motion 1 computed from two different pairs of cameras. (h) shows those on motion 2.

from motion 2. Note that the motions of these 4 points are non-rigid. The 4 points with 2 frames provide us 8 points in the space-time. Thus we chose 6 points from these 8 points, and computed space-time invariants from (4). The space-time invariants computed from motion 1 and motion 2 are shown in Fig. 7 (g) and (h) respectively. The red and blue curves in these figures show the space-time invariants computed from two different pairs of cameras. As shown in these figures, the space-time invariants are almost identical if the non-rigid 3D motions of multiple points are same, and the invariants are different if the 3D motions are different. Thus, even if the viewpoints are arbitrary, we can distinguish non-rigid 3D motions efficiently by using the proposed space-time invariants.

8 Conclusions

In this paper, we proposed a method for computing projective space-time invariants for distinguishing non-rigid 3D motions from arbitrary viewpoints. We first proposed a method for computing projective space-time invariants for non-rigid 3D motions. In particular, we showed that projective space-time invariants for 3D motions can be computed from 6 points in the 4D space-time. We also showed that even if the space-time projection under perspective cameras cannot be described by affine projections nor projective projections, we can still compute invariants on 3D motions by using projective camera calibrations. We next showed that we can further stabilize the projective space-time invariants by using the mutual projection of multiple cameras. The proposed space-time invariants were tested by using real image sequences taken from 3D motions. Since the projective space-time invariants are identical even if the viewpoints are different, the result of the proposed method can be used for distinguishing 3D motions from video sequences captured at arbitrary viewpoints.

References

1. A.F. Bobick and A.D. Wilson, *A state-based technique for the summarization and recognition of gesture.* In Proc, 5th International Conference on Computer Vision, pages 382-388, Cambridge, USA, 1995.
2. T. Darrell and A. Pentland, *Space-time gestures.* In Proc. Conference on Computer Vision and Pattern Recognition, pages 335-340, New York, 1993.
3. R.I. Hartley, *In defense of the eight-point algorithm.* IEEE Trans. Pattern Analysis and Machine Intelligence, 19(6):580-593, 1997.
4. I. Laptev and T. Lindeberg, *Space-time interest points* . In Proc. 9th International Conference on Computer Vision, volume 1, pages 432-439, 2003.
5. A. Levin, L. Wolf and A. Shashua, *Time-Varying Shape Tensors for Scenes with Multiply Moving Points.* In Proc. Conference on Computer Vision and Pattern Recognition, 2001.
6. D. Liebowitz and S. Carlsson, *Uncalibrated Motion Capture Exploiting Articulated Structure Constraints.* In Proc. 8th International Conference on Computer Vision, volume 2, pages 230-237, 2001.
7. J.L. Munday and A. Zisserman, *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, USA, 1992.
8. J. Sato, *Space-time invariants and recognition of motions from arbitrary viewpoints.* Trans. Institute of Electronics Information and Communication Engineers, J84-D-II (8): 1790-1799, 2001.
9. H. Ohno and M. Yamamoto, *Gesture Recognition using Character Recognition Techniques on Two-dimensional Eigenspace* . In Proc. 7th International Conference on Computer Vision, volume 1,pages 151-156, 1999.
10. L. Quan, *Invariants of six points and projective reconstruction from three uncalibrated images.* IEEE Trans. Pattern Analysis and Machine Intelligence, 17(1):34-46, 1995.
11. I. Weiss, Geometric invariants and object recognition. *International Journal of Computer Vision*, Vol. 10, No. 3, pp. 207- 231, 1993.
12. Z. Zhang, *Determining the epipolar geometry and its uncertainty: A review*, In Proc. International Journal of Computer Vision, 27(2):161-195, 1998.
13. A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow, 3D object recognition using invariance. *Artificial Intelligence*, Vol. 78, pp. 239-288, 1995.

Detecting and Tracking Distant Objects at Night Based on Human Visual System

Kaiqi Huang, Liangsheng Wang, and Tieniu Tan

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, P.R. China, 100080
{kqhuang, lswang, tnt}@nlpr.ia.ac.cn

Abstract. Moving object detection is a challenging task for night security because of bad video quality. In this paper, we propose a robust real time objects detection method for night visual surveillance based on human visual system. By measuring contrast information variation in multiple successive frames, a spatio-temporal contrast change image *(CCI)* is formed. Then the multi-frame correspondence technology is employed to robustly extract salient motions or moving objects from *CCI*. Since *CCI* is a statistical measurement of variation based on human visual system, the proposed method is effective at night and better than traditional detection methods. Experiments on real scene show that the method based on contrast feature is effective for night object detection and tracking, our approach is also robust to camera scale variation as well as low computation cost.

1 Introduction

Night security has gradually attracted more and more attentions. Object detecting and tracking is the first step and have been studied widely. Various approaches have been proposed including feature-based object detection [1,2]、template-based object detection [6,7] and background subtraction or inter-frame difference based detection [3,4,5].

However, most of these approach intent to solve object detection on daytime. As we know, nighttime image or video captured by common CCD camera has low brightness, low contrast, low Signal to Noise Ratio (SNR) and nearly no color information, so it is a great challenge to detect objects at night because most of features used in daytime such as color, local edge, contour features etc. will fail at night. It is also very difficult to model night background because of noise and variation of lighting condition. [1]While human can easily detect object in this condition and human visual system characteristics have been proved useful in computer vision and image processing field, such as image enhancement, denoising, compression and watermarking applications [8,9]. In this paper, motivated by human visual system, we will propose a robust object detection and tracking method at night. Contrast, as one of the most important feature to distinguish objects from background for human, is used to

[1] Here, we focus on the light condition where human can discriminate objects, otherwise thermal Infrared camera will be considered, which is used in night visual surveillance for objects detection while the cost is so high that it is not be considered in this condition [10].

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 822 – 831, 2006.

detect object in the first step, then temporal information (contrast change information) is combined to get more robust results. Experiments on real night scene show that the contrast feature is effective for night object detection and tracking and our approach is effective for camera zooming objects detection and tracking as well as low computation cost.

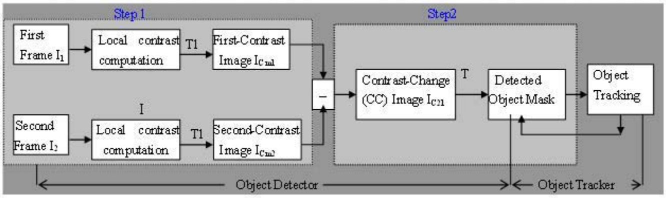
2 Algorithm Motivation

There are evidences from psychology physiology show that response of the Human Visual System (HVS) depends much less on the absolute luminance than on the relation of its local variations to the surrounding background, which is measured by contrast and is commonly used in vision models [8,9]. Human can detect objects only if the contrast is above some threshold. Coarseness is another significant feature for giving information about the size of the objects, which is described by various windows. The higher the coarseness value is, the rougher the object is [18]. On the other hand, human pay more attention to change and motion provides other useful information for objects detection and tracking [15]. Here we use local contrast to describe coarseness and contrast and contrast change (CC) to describe the motion information. Based on the human visual system characteristics, we give the algorithm framework in the next section.

3 Object Detection and Tracking Algorithm for Night Visual Surveillance Based on HSV

3.1 Algorithm Framework

Our algorithm framework consists of object detection and tracking as Fig 1. Similar to human discrimination model, the object detection algorithm includes two steps: visible content detection based on local contrast computation and moving object detection based on contrast change (CC). In the first step, the visible object will be detected based on local contrast computation and we will get the contrast images I_{Ci}. In this step, all the contents we can see can be detected, including the interesting objects or not. Contrast Change (I_{Cij}), which gives information of moving objects in last frame and current frame, is used to get the moving objects in the second step. i、j means


Step1
Step2
First Frame I_1
Local contrast computation
T1
First-Contrast Image I_{Cm1}
Second Frame I_2
I
Local contrast computation
T1
Second-Contrast Image I_{Cm2}
−
Contrast-Change (CC) Image I_{C21}
T
Detected Object Mask
Object Tracking
Object Detector
Object Tracker

.

Fig. 1. Algorithm framework of object detection and tracking for night visual surveillance

frames. Object tracking will not only get the correspondence between objects but also filter the false detection objects by feedback. Next we will give the algorithm details.

3.2 Visible Image Content Detection Based on Local Contrast Computation

Contrast plays an important role to detect object for human especially at night because most of the features used by present detection algorithm such as color, contour, local edge are lost. Based on this motivation, we propose local contrast feature to detect objects. There are many methods to compute contrast [11,12,13,14]. Typically, luminance contrast is defined as the relative difference between luminance of the object, L_o, and the surrounding background, L_B, as $C = {(L_o - L_B)}/{L_B}$, which is called Weber contrast [13]. Michaelson defined contrast for elementary patterns as $C = {(L_{max} - L_{min})}/{(L_{min} + L_{max})}$ [14]. Recently more complex contrast computation methods are proposed in the FFT and wavelet domain [11, 12].Here, we make use of a simple and low computation statistics contrast method, defined as the standard deviation σ of all pixel intensities divided by the mean intensity μ [16],

$$C = {\sigma}/{\mu} \tag{1}$$

Considering the coarseness feature, we can compute the local contrast $C_{(p,q)} = {\sigma_{(p,q)}}/{\mu_{(p,q)}}$, where [p,q] is the window size.

Table 1 gives some local contrast computation examples. The window size for local contrast computation is 44×50. As Table 1, we can detect one person from frame 75-100, most of the contrast of frame 75-100 is larger than some threshold (nearly 0.6), and the contrast of black part is only 0.4280, there is distinguished difference between this two kinds of image. It is should be mentioned that the contrast of light part is also over 0.6 (equal to 0.6335), which shows that the edge part will also can be detected in the result of contrast image but it will be stable most of time and can be filtered by contrast change step.

From analysis of Table 1, we can get formula to compute local contrast image I_{Cm} as

$$I_{C_m} = T1[I(x_{p,q}, y_{p,q})] \tag{2}$$

Table 1. Local contrast computation for three kinds of image

Images[44×50]								
FrameNumber	·75	80	85	90	95	100	Black-part	Light-part
σ (Std.)	0.1968	0.2064	0.1817	0.1713	0.1993	0.1731	0.1148	0.3013
μ (Mean)	0.3251	0.3070	0.3566	0.2895	0.2778	0.2767	0.2681	0.4755
C(Contrast)	0.6052	0.6725	0.5095	0.5919	0.7032	0.6258	0.4280	0.6335

Where contrast threshold $T1$ [2]indicates we can see content above the threshold and $I(x_{p,q}, y_{p,q})$ is the image after local contrast computation. Fig 2 is another example about visible objects detection step on real scene night image, the window size is 16x20 and threshold $T1$ is 0.45. (a) is original image with the size of 320x240, (b) is the detection result. The red rectangles indicate the visible content in the image. We can see that the interesting object and other information are also detected by this step.

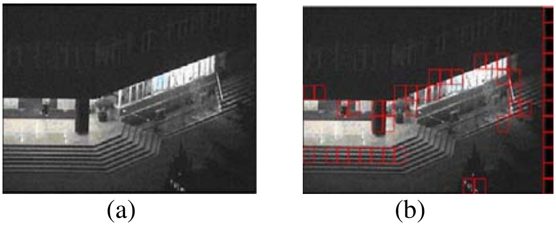

(a) (b)

Fig. 2. (a) original image, (b) visible content

3.3 Moving Object Detection Based on Contrast Change

In the first step, we have detected the visible image content including the interesting objects and some information with high edge. Human often pays more interesting to the change, so in the second step we will use the contrast change (CC) to filter the result in the first step and get the moving objects. This step can be indicated by Formula (3) as followed:

$$I_{C_{21}} = T(\left|I_{C_1} - I_{C_2}\right|) \tag{3}$$

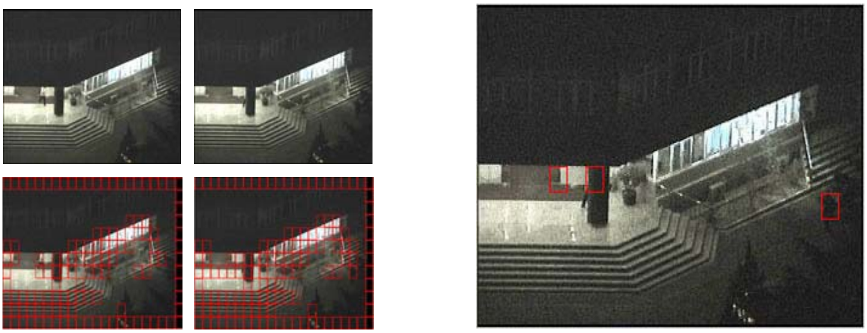
Where I_{C_1} and I_{C_2} are contrast images we get from the first step, $I_{C_{21}}$ is the contrast difference between two contrast images. T is the threshold to filter some little change caused by other factors such as noise or little light variation.

Fig 3 is an example on real scene video captured by CCD camera at night. (a) shows local contrast computation results of two frames. We can see that most of the visible content in the images can be detected by this step. (b) is the result based on contrast change. We plot the detection result on the second frame. The three red rectangles indicate the objects contrast change between two frames and locate the objects position in the last and current frame. The result is so good as to detect the people running from dark place and people occluded by pillar.

3.4 Object Tracking

There are a large number of object tracking methods, which can be classified into four groups: region-based, contour and mesh-based, model-based and feature-based [17]. As we have detected the rectangle region, we focus on region-based tracking method here.

[2] $T1$ indicates we can see content above the threshold, and threshold T in the next step indicates the change we can detect above the threshold ,threshold T ,which can be decided by p tile method adaptively in every frame.



(a) Visible image content detection based on contrast

(b) Moving object detection based on Contrast Change (CC)

Fig. 3. Objects detection results based on contrast and contrast change

In the last step, objects may be detected by several local windows, such as Fig. 4.(a), so we should choose a bounding box including the local windows as Fig. 4(b), which can be solved by grouping together 8-connected clusters of blocks having similar (and non-zero) displacement vectors [14]. Once the object areas are determined in each frame, the tracking algorithm is needed to trace the objects from frame to frame.

Fig. 4. (a) connected before (b) connected behind

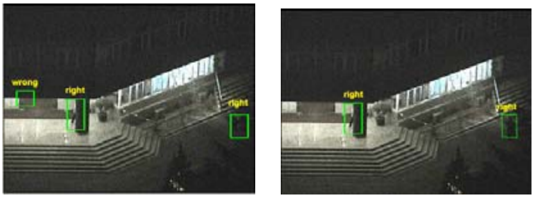
The tracking algorithm includes two purposes:

To establish correspond relationship by distance computation between the last frame and current frame.

The result of the distance computation can be represented as a matrix $D=\{d_p,d_q\}$, where each row, p, corresponds to a rectangle descriptor in frame n+1, and each column, q, corresponds to a rectangle descriptor in frame n.We refer to this matrix as distance matrix. Each element of the distance matrix represents the distance between two rectangles. The element under some threshold for each row and for each column identifies a possible correspondence between two rectangles.

To filter the false detection objects as Fig 5(a) caused by edge or large light variation by multi-frame matching relation.

When a rectangle region is detected while there is no corresponding object in the past, it may signal a candidate. If the candidate can be traced successfully for several frames (here two frames), it will then be considered as a new object and displayed with a unique label assigned, otherwise it will be discarded. By this way, we can get a more accurate detection result as Fig.5. (b).



(a) (b)

(a) false object detection caused by large light variation;

(b) false object has been filtered by temporal duration.

Fig. 5. Example for discarding the wrong detection object

4 Experimental Results and Discussion

In this section, the results of proposed algorithm for real scene night objects detection and tracking are assessed. All the real scene night videos are captured by common CCD camera with the sizes of 320x240. In our experiments, only value in R channel is used as the input. The effectiveness of the proposed method is verified by detection results and tracking results.

4.1 Algorithm Testing

First, we test the detection algorithm from frame 1 to frame 120 for sequence "*two person*". Fig.6 gives the detection results from frame 1-120. The first column is the

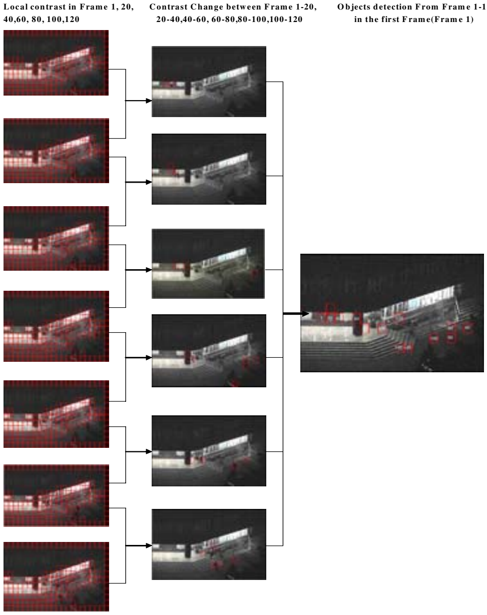
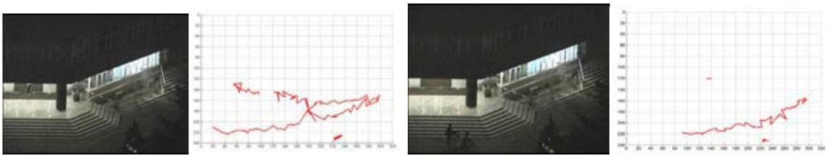

Local contrast in Frame 1, 20, 40,60, 80, 100,120
Contrast Change between Frame 1-20, 20-40,40-60, 60-80,80-100,100-120
Objects detection From Frame 1-120 in the first Frame(Frame 1)

Fig. 6. Detection results from Frame 1-120, first column: Local Contrast computation results. Second column: Contrast change results From Frame 1-120. Third column: Detection results from Frame 1-120 on the first frame (frame1).



(a) (b)

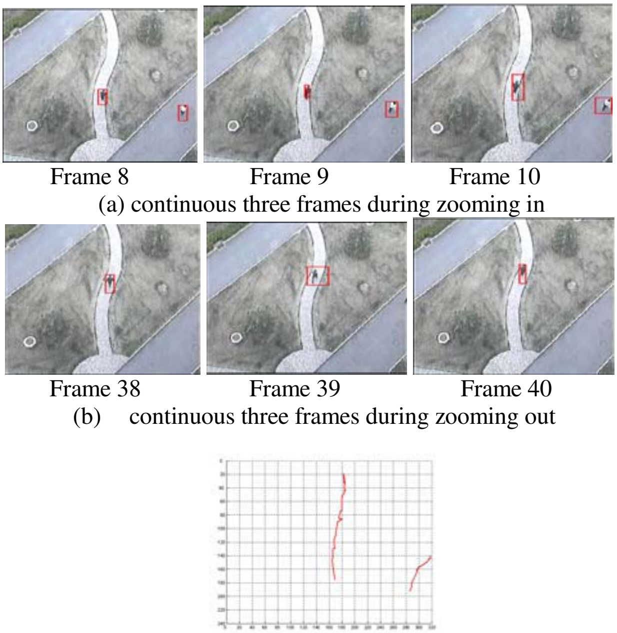
(a) First frame from test sequences "t*wo persons", "bikemen",*

(b) Trajectories of video objects for the sequence in the corresponding row.

Fig. 7. Trajectories of real scene night objects. The horizontal and vertical axes of the graphs represent the width and the height of the frame, respectively.

local contrast computation result in frame 1, 20, 40, 60, 80, 100, 120. As we can see, the moving objects and other visible content can be detected in this step. The second column is the contrast change result between frame1, 20, 40, 60, 80, 100, 120. As we have hoped, the results are good to detect the change. We plot all the detection results from frame 1 to frame 120 on the first frame as the third column in Figure 6, It is clear that all the positions the objects appear can be detected accurately from frame 1-120.

Fig 7 shows the tracking results from two sequences "*two persons*" and "*bike-man*". The left column as (a) is the first frames of the two sequences, (b) gives the trajectories of objects for the sequences in the corresponding row. It is clear that the two persons in the first sequence can be detected and tracked exactly except for



Frame 8 Frame 9 Frame 10

(a) continuous three frames during zooming in

Frame 38 Frame 39 Frame 40

(b) continuous three frames during zooming out

(c) Trajectories of camera zooming in and zooming out

Fig. 8. Objects detection and tracking for camera zooming

the occlusion by pillar. The occlusion tracking is not the focus in this paper, more information for occlusion tracking should be considered in the future. In the second sequence, two persons are also can be robustly detected and tracked while they are so close that they are detected as one object and there is one trajectories in the right.

Second, our algorithm is also robust to camera zooming, which outgoes background modeling based detection methods in substance. Fig 8 (a) is the detection results of continuous three frames for camera zooming in and (b) is the detection results of continuous three frames for camera zooming out. We can see that the objects can be detected accurately without losing. (c) is the trajectories of objects when camera zooms in and zooms out. It is clear that objects can be tracked robustly without losing. All video results can refer to the attachments.

4.2 Algorithm Evaluation

We also give the comparison between our algorithm and Adaptive Mixture Gaussian Model (MOG) based method [3] by the measure similar to Jaccard coefficient [19], which gives the detection accuracy for each frame of sequence:

$$J = {}^{TP}\!\big/_{(TP + FP + FN)} \tag{4}$$

Where True positives (TP) is number of moving objects correctly detected. False positives (FP) is number of false detection by the algorithm. False negatives (FN) is number of missing detection objects by the algorithm. Detection Ground truth is given by our eyes for each frame of the sequence.

Fig 9 is the detection result comparison between our algorithm and MOG based method, which is classic as one of the most popular object detection algorithm, (a) is the detection results of CC with tracking feedback, (c) is the detection results of CC without tracking feedback, (b) and (d) are detection results of MOG with different variance. The horizontal axes are frame index and the vertical axes are Jaccard coefficient. Higher J coefficient indicates the better detection results. We can see that J is the best for most frame of (a) and (c) is the worst. For MOG algorithm, (b) gets the better result in missing object detection (frame 60-100) while worse in false object detection in the beginning and (d) gets the inverse result, but both of them will be better after some frames learning (from frame 100).

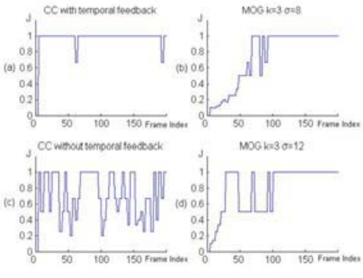
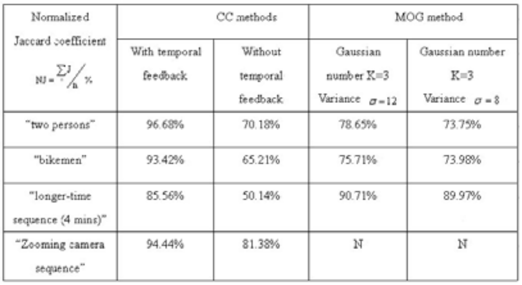


Fig. 9. Algorithm comparison for each frame of sequence "two persons"

Table 2. Algorithm comparison with MOG method on difference sequences



Normalized Jaccard coefficient $NJ = \sum J / n \%$	CC methods		MOG method	
	With temporal feedback	Without temporal feedback	Gaussian number K=3 Variance $\sigma = 12$	Gaussian number K=3 Variance $\sigma = 8$
"two persons"	96.68%	70.18%	78.65%	73.75%
"bikemen"	93.42%	65.21%	75.71%	73.98%
"longer-time sequence (4 mins)"	85.56%	50.14%	90.71%	89.97%
"Zooming camera sequence"	94.44%	81.38%	N	N

Table 2 gives the comparison between our algorithm and MOG based methods on different sequences. The results of CC method and MOG methods are compared by Normalized Jaccard Coefficient, which gives the detection accuracy for whole sequence. For sequence "two persons" and "bikemen", it is clear that the result of CC method with tracking feedback is the best. As we have tested in Fig 9, the MOG methods with parameter $\sigma = 12$ does well for false detection(noise compression) while not good for missing detection and MOG methods with parameter $\sigma = 8$ achieves inverse result, so these NJ are neither high as our method. We also test these algorithms with about 4-minutes sequence at the same scene as above two sequences, the results of MOG method are slightly more better for either $\sigma = 12$ or $\sigma = 8$ in this condition because MOG method has the learning ability, but for zooming camera sequence, it will not work while our method is robust to it.

5 Conclusion and Future Work

In this paper, motivated by human vision system, we proposed objects detection and tracking algorithm for night visual surveillance. The objects detection is based on local contrast saliency information and detection results can be improved by tracking step. Experimental results have demonstrated that our approach has the ability to detect and track objects robustly at night as well as camera scale changing, which is challenging for most of present methods. In the future, more information should be considered for occlusion tracking at night.

Acknowledgement

The work reported in this paper was funded by research grants from the National Basic Research Program of China (No. 2004CB318100) and the National Natural Science Foundation of China (No. 60335010),(No. 2004DFA06900).

References

1. D. Crandall, J. Luo. "Robust color object detection using spatial-color joint probability functions," CVPR (2004). 379-385
2. C.G. Harris and M. Stephens, "A combined corner and edge detector," in 4th Alvey Vision Conference, (1988)147–151.

3. C. Stauffer, Grimson,W.E.L. “Adaptive background mixture models for real-time tracking”. CVPR, (1999).252-260,
4. C.R. Wren, A. Azarbayejani, and A. Pentland, “Pfinder: realtime tracking of the human body,” PAMI, v. (1997).780–785,
5. L.Liyuan, Weimin Huang, et.al “Statistical modeling of complex backgrounds for foreground object detection” IEEE Trans. on Image Processing, V13 (11), (2004)1459 - 1472
6. A.K.Jain,Y.Zhong and S.Lakshmanan, “Object matching using deformable templates”. PAMI v.8(3). (1996).267-278,
7. S. Sclaroff, Lifeng Liu, “Deformable shape detection and description via model based region grouping”. PAMI, (2001)474-489,
8. [8-9] Anonymous
10. D. Davies, P. L. Palmer, M. Mirmehdi, “Detection and tracking of very small low contrast objects.” BMVC, (1998)599-608,.
11. J.Tang, Kim JH, Peli E. “Image enhancement in the JPEG domain for people with vision impairment.” IEEE Transactions on Biomedical Engineering; 51(11): (2004)2013-2023,
12. E.Peli, “Contrast sensitivity function and image discrimination”. J Opt Soc Am A, 18(2): (2001) 283-293 .
13. R. M. Haralick, L. G. Shapiro, Computer and Robot Vision, Addison-Wesley, Vol.1, (1992)28-33,
14. P. G. J. Barten, Contrast sensitivity of the Human Eye and Its Effects on Image Quality, SPIE, Bellingham, Washington, (1999).
15. K. Gould, K. Rangarajan, and M. Shah, “Detection and representation of events in motion trajectories,” chapter in Advances in Image Processing and Analysis, (editors: Gonzalez and Mahdavieh), Optical Engineering Press, (1992)
16. E. Reinhard, Peter Shirley, Michael Ashikhmin and Tom Troscianko, “Second order mage statistics for computer graphics"” ACM Symposium on Applied Perception in Computer Graphics and Visualization, August (2004).
17. A. Cavallaro, O. Steiger, T. Ebrahimi, "Multiple video object tracking in complex scenes", Proc. of ACM Multimedia, Juan les Pins, France, (2002)1-6.
18. H. Tamura, S. Mori, and T. Yamawaki, “Texture features corresponding to visual perception”. IEEE Trans. on Syst., Man, and Cybern., (1979)6 (4), 460-473,.
19. P. Sneath,, R. Sokal, ,. Numerical Taxonomy. The principle and practice of numerical classification. W.H.Freeman. (1973)

Motion Guided Video Sequence Synchronization

Daniel Wedge*, Du Huynh, and Peter Kovesi

School of Computer Science & Software Engineering,
The University of Western Australia,
Crawley, W.A., Australia
{dwedge, du, pk}@csse.uwa.edu.au

Abstract. We present an algorithm that synchronizes two short video sequences where an object undergoes ballistic motion against stationary scene points. The object's motion and epipolar geometry are exploited to guide the algorithm to the correct synchronization in an iterative manner. Our algorithm accurately synchronizes videos recorded at different frame rates, and takes few iterations to converge to sub-frame accuracy. We use synthetic data to analyze our algorithm's accuracy under the influence of noise. We demonstrate that it accurately synchronizes real video sequences, and evaluate its performance against manual synchronization.

1 Introduction

An increasing number of computer vision applications are being developed that process multiple videos recorded simultaneously from different locations. Some applications of multiple view video analysis include comparisons of human motion [1], virtualized reality [2] and reconstruction of non-rigid scenes [3]. In these applications, synchronization is essential to ensure consistency in the structure recovered from the videos.

Synchronization involves finding the temporal relationship between two or more video sequences. Most literature focuses on a linear model, where there is a temporal offset Δ between the sequences, and the ratio of frame rates is denoted α. This can be expressed mathematically by:

$$j = \alpha i + \Delta, \tag{1}$$

where i and j are frames from each sequence recorded at the same instant in time.

Synchronization can be performed in hardware, for example, by embedding a timestamp in the video stream, or sending a synchronization signal to cameras [2]. However, this can be costly and must be set up prior to recording. Alternatively, software algorithms can recover synchronization from visual cues.

There are two general classes of synchronization algorithms: direct and feature-based alignment. Direct alignment [4] uses all pixels in a video frame for synchronization and is suitable for videos containing lighting changes, e.g., fireworks. Feature-based alignment uses features such as points on moving objects or object trajectories as a basis for the synchronization algorithm.

Many feature-based synchronization methods are based on a multiple-view geometric entity such as the fundamental matrix or homography. When either entity is estimated from stationary background scene points, the synchronization can be established

* Daniel Wedge wishes to acknowledge the support of an Australian Postgraudate Award.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 832–841, 2006.

via the calculation of reprojection errors of moving object points [5]. For instance, Reid and Zisserman [6] synchronize two videos of a critical passage of play in a soccer match by firstly relating the homography of the ground plane between two views using ground markings; they then align the sequences by minimizing the reprojection errors of players' shadows on the ground. Similarly, with object motion occurring mainly on the dominant ground plane, Stein [7] and Lee et al. [8] synchronize multiple surveillance videos using homographies. A fundamental matrix based algorithm proposed by Pooley et al. [9] synchronizes sequences captured by two moving cameras via the Hough transform on a reparameterized space of α and Δ. They refine these estimated parameters using a gradient descent method to minimize the reprojection error. A further fundamental matrix based algorithm developed by Carceroni et al. [10] employs the epipolar constraint to establish tentative synchronized frames to estimate an N-dimensional timeline using RANSAC [11]. The timeline encapsulates the ratio of frame rates and temporal misalignment between all pairs of the N video sequences to be synchronized.

A feature-based algorithm by Caspi and Irani [4] synchronizes two sequences by integrating multiple trajectory observations where α is known, and the cameras have fixed internal parameters throughout the sequences. A homography or fundamental matrix between views is also estimated, depending on the type of motion contained in the scene (planar for a homography, or free motion for a fundamental matrix). An iterative step alternately refines the temporal offset, and the homography or fundamental matrix.

Tuytelaars and Van Gool [12] track five points undergoing non-rigid motion throughout each video sequence, and recover an affine projection matrix for each camera. In each frame, one tracked point is back-projected into space, and for temporally corresponding frames, lines corresponding to the same 3D point will intersect; this is the basis of their synchronization algorithm.

Wolf and Zomet [13] use the singular values of the joint image measurement matrix to synchronize videos recorded by two affine cameras. Their algorithm does not require the specification of stationary background points, nor point correspondences on moving objects in each sequence. Tresadern and Reid [3] extended this method to synchronize to sub-frame accuracy, where the cameras film at different frame rates; however, their algorithm requires objects to be tracked through a video sequence, and corresponding trajectories in each sequence to be known. Giese and Poggio [14] tackled the modelling of biological motion patterns as a synchronization problem. Later, Rao et al. [1] solved for a non-linear temporal relationship between two sequences. Their algorithm synchronizes videos of the same action performed at different rates, e.g., dancing routines. The smallest singular value of the measurement matrix used in the linear estimation of the fundamental matrix is used as an error measure.

An alternative feature-based approach is demonstrated by Yan and Pollefeys [15], who analyze the distribution of space-time interest points [16] throughout two video sequences to solve for Δ. The curves that represent the distribution of space-time features throughout each sequence are cross-correlated for a range of frame offsets; the offset where the maximum cross-correlation score is achieved is deemed the actual alignment.

Frontier points are an alternative feature used by Sinha and Pollefeys [17] for synchronization of videos containing object silhouettes recorded by cameras operating at the same frame rate. A RANSAC based approach considers lines tangential to the silhouette's convex hull and passing through frontier points as potential epipolar lines. The algorithm can simultaneously estimate the temporal offset and the fundamental matrix.

We introduce an algorithm that synchronizes two short videos of an object undergoing ballistic motion, recorded using stationary cameras with fixed intrinsic parameters. We use stationary background points to estimate a fundamental matrix, and exploit the motion of an object moving in a short ballistic trajectory to rapidly converge to the correct synchronization. Our method is based on epipolar geometry and can accurately synchronize videos recorded with both an unknown temporal offset and ratio of frame rates. Our work is similar to Carceroni et al. [10] in that epipolar geometry is used to find corresponding frames; however, we exploit object motion to converge to the correct solution. We present results where the algorithm is applied to synthetic and real data where the cameras remain stationary. We show that the algorithm can synchronize sequences to sub-frame accuracy, and that the influence of noise on the coordinates in the trajectory used for synchronization is not significant.

2 Motion Guided Synchronization

The algorithm consists of three steps: estimating the fundamental matrix from a number of corresponding stationary background points; finding temporally corresponding points on the object's trajectory in each view iteratively by exploiting object motion and epipolar geometry; lastly, estimating the ratio of frame rates and the frame offset from pairs of temporally corresponding frames.

In the following sections, frame i in video sequence 1 is referred to as $\mathcal{S}_i$, and frame j in sequence 2 is denoted by $\mathcal{S}'_j$. Image points and lines are in homogeneous coordinates, and denoted by lowercase boldface letters, e.g., $\mathbf{x}_i$ denotes an image point in frame $\mathcal{S}_i$. Points and lines in sequence 2 are distinguished by a prime, e.g., $\mathbf{l}'_j$. Throughout this paper, we use the term *corresponding frames* to mean a pair of frames, one from each video sequence, that are recorded at the same time instant. We use a ball as the moving object in this paper, though it can be substituted for any object undergoing ballistic motion. To avoid confusion with other moving objects in the videos, we refer to the moving object as a ball throughout this paper.

2.1 Finding Corresponding Frame Pairs

Given that the position of the moving ball has already been identified by a feature tracking process, we aim to establish temporal correspondences of the ball's motion in the two video sequences by exploiting epipolar geometry and the ball's motion.

In our algorithm, the fundamental matrix $\mathtt{F}$ [5] is estimated from stationary background points. If the ball's location $\mathbf{x}_i$ is known in frame $\mathcal{S}_i$, we can compute the corresponding epipolar line $\mathbf{l}'_i$ in $\mathcal{S}'_j$, via $\mathbf{l}'_i = \mathtt{F}\mathbf{x}_i$. For video sequences recorded by stationary cameras with fixed intrinsic parameters, $\mathtt{F}$ is invariant and the images of all corresponding stationary scene points satisfy the epipolar constraint. If $\mathcal{S}_i$ and $\mathcal{S}'_j$ are corresponding frames, then the ball's imaged position $\mathbf{x}'_j$ in $\mathcal{S}'_j$ will lie on $\mathbf{l}'_i$, and the epipolar constraint holds. This fact is used to search for the correct alignment.

First, a frame $\mathcal{S}_i$ is randomly chosen where the ball's vertical velocity is significant (explained later in Section 4). Rather than searching through all values of j to find the corresponding frame, the ball's motion is exploited to reduce the search. A frame $\mathcal{S}'_j$ is selected such that the direction of the ball's vertical motion is the same as in frame $\mathcal{S}_i$. It is assumed that the frame rate is sufficiently high that the ball's inter-frame motion is approximately linear. Thus, the ball's velocity in $\mathcal{S}'_j$ can be approximated linearly from the ball locations in two consecutive frames. Then, from the ball's position in $\mathcal{S}'_j$ and its velocity, the number of frames until the ball crosses the epipolar line $\mathbf{l}'_i$ can be calculated. The following steps outline an iterative method to estimate the value of j given an epipolar line $\mathbf{l}'_i$:

1. Calculate the ball's inter-frame velocity: $\mathbf{v}'_j = \mathbf{x}'_{j+1} - \mathbf{x}'_j$.
2. Let $\mathbf{t}'$ be the linear approximation of ball's trajectory in $\mathcal{S}'_j$. Assuming a constant velocity model, $\mathbf{t}'$ is a straight line passing through $\mathbf{x}'_j$ and $\mathbf{x}'_{j+1}$, calculated via the cross product: $\mathbf{t}' = \mathbf{x}'_j \times \mathbf{x}'_{j+1}$.
3. Next, calculate the intersection point $\mathbf{p}'$ of the approximated trajectory $\mathbf{t}'$ and the epipolar line $\mathbf{l}'_i$: $\mathbf{p}' = \mathbf{t}' \times \mathbf{l}'_i$.
4. To estimate the number of frames until the ball crosses the epipolar line, firstly let $\mathbf{d}'$ be the vector from $\mathbf{x}'_j$ to $\mathbf{p}'$. Then the ball is estimated to cross the epipolar line in $n = \|\mathbf{d}'\|/\|\mathbf{v}'_j\|$ frames. However, there is an ambiguity in the motion of the ball as the ball may have already passed the intersection point, in which case the algorithm must look backwards in time .
5. The direction ambiguity can be resolved by examining the vectors $\mathbf{v}'_j$ and $\mathbf{d}'$. The search should be directed forwards if both $\mathbf{v}'_j$ and $\mathbf{d}'$ have the same orientation, otherwise, the search should move backwards. n is then modified: $n \leftarrow n\,\mathrm{sgn}(\mathbf{v}_{\mathrm{j}}'^{\top}\mathbf{d}')$.
6. - If $n \in [0, 1)$ then the ball must have just crossed the epipolar line in the time interval $[j, j+1)$. The synchronized frame can then be estimated to be frame $j \leftarrow j + n$. At this stage, the iteration can terminate and the temporal correspondence $\mathcal{S}_i \leftrightarrow \mathcal{S}'_j$ is established.
   - If $n \notin [0, 1)$ then the frame that is closest in time to frame i must be $j + \lfloor n \rfloor$, where $\lfloor n \rfloor$ is the largest integer less than n. The strategy is to update j as $j \leftarrow j + \lfloor n \rfloor$ and repeat the process by looping back to Step 1. Although j is updated by a whole number here, synchronization to sub-frame accuracy is achieved when the iteration terminates as described in the previous paragraph. We note that an integer update $\lfloor n \rfloor$ is enforced because the ball's velocity is approximated using the forward, rather than the backward, difference.

2.2 Convergence of Algorithm

A formal proof of the convergence of the algorithm in the previous subsection is not given here except for a brief mention of the idea and an example. Firstly, we divide the algorithm into two cases: case (a), where the ball is above the epipolar line $\mathbf{l}'_i$; and case (b), where the ball is below $\mathbf{l}'_i$. In case (a), the value of n will be overestimated because the magnitude of the ball's vertical velocity increases as it approaches $\mathbf{l}'_i$, and this apparent acceleration causes the distance to $\mathbf{l}'_i$ to be covered in fewer frames than is estimated under a constant velocity model. In case (b), using a constant velocity model causes n to be underestimated when the ball is below the epipolar line, for the same reason as in case (a). Because of this underestimation, if the algorithm starts in case (a),

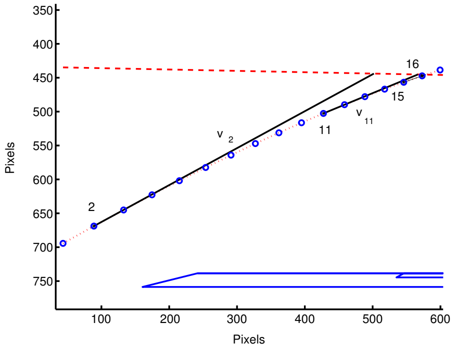


Fig. 1. An example illustrating the convergence of our algorithm, as detailed in Section 2.2

it will soon reach case (b) because of the overestimation. Then, n will converge to 0 as in case (b), the algorithm underestimates the number of frames, and at each step, the distance from the ball to the epipolar line always decreases and never increases.

Fig. 1 shows an example where the ball is moving upwards towards the epipolar line and decelerating, with the iteration arbitrarily starting at $j = 2$. The velocity of the ball is shown by the solid line v_2 and, with the constant velocity assumption, is expected to cross the epipolar line (dashed) in frame 11. This estimate is refined to frame 15 using the ball's velocity v_{11} at frame 11, then to frame 16, and finally, frame 16.20. This example demonstrates n being underestimated in case (b).

2.3 Combining Observations

Given a frame $\mathcal{S}_i$, the iterative procedure in Section 2.1 estimates the corresponding frame j in $\mathcal{S}'$ to sub-frame accuracy. If this procedure is repeated for frames $\{i_1,\ldots,i_m\}$, for $m \geq 2$, then the set of corresponding frames $\{j_1,\ldots,j_m\}$ from the second video sequence can be established. Since the values $\{i_1,\ldots,i_m\}$ and $\{j_1,\ldots,j_m\}$ are related via Equation (1), the values of α and $\mathit{\Delta}$ can be estimated via least-squares.

3 Results

We present results for synchronizing synthetic and real data sets. The synthetic data allow us to test the accuracy of our algorithm in the presence of added noise, whilst testing on real videos shows that the algorithm can be applied in a practical setting.

We used simulated trajectories to analyze the accuracy of our algorithm in the presence of noise. A large number of these tests showed that our algorithm can accurately synchronize two videos containing only one trajectory; on average, the frame rate ratio was accurate to within 2% of the actual ratio, and the estimated frame offset shown to have a small mean error, particularly when many pairs of corresponding frames are used to recover α and $\mathit{\Delta}$. The setup used in our experiments on synthetic data is shown in Fig. 2. The ball's maximum vertical speed was 26 pixels/frame at the point of the trajectory closest to the camera, and averaged 7 pixels/frame throughout both sequences.

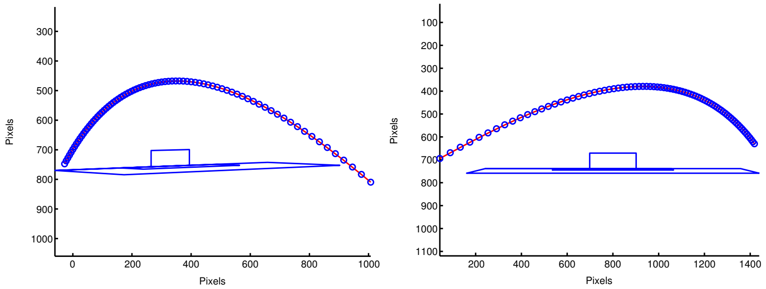


Fig. 2. A trajectory and football goals as viewed by two cameras, with the ball locations in each frame marked by circles

In the following subsections, we analyze the effect of adding noise to the trajectories in both sequences, and also to the stationary points used to estimate the fundamental matrix. We also demonstrate the application of the algorithm to real video sequences and compare the recovered synchronization parameters with manual synchronization. We use $\bar{\alpha}$ and $\widehat{\alpha}$ to denote the true and recovered values of α respectively. Similarly, we use $\bar{\Delta}$ and $\widehat{\Delta}$ to indicate the real and estimated frame offsets.

3.1 Simulated Experiments

We conducted experiments to evaluate the accuracy of the algorithm in the presence of isotropic Gaussian noise. The noise was added to the x and y components of the ball's position, and no smoothing process was applied to the trajectory. Table 1 shows the results for integrating different numbers of corresponding frames for various standard deviations, σ, of Gaussian noise. In these experiments, the ratio of frame rates and the frame offsets were fixed to isolate the influence of noise. The true values were $\bar{\alpha} = 5/6$ and $\bar{\Delta} = 3.5$ frames, and the sequences contained 80 frames. At each level of noise, we compared the results of using 2, 5, and 8 pairs of corresponding frames in estimating $\widehat{\alpha}$ and $\widehat{\Delta}$.

It can be seen from Table 1 that when noise is present, the accuracy of $\widehat{\alpha}$ and $\widehat{\Delta}$ is improved by solving for more pairs of corresponding frames. Even with significant amounts of noise, the algorithm accurately computes $\widehat{\alpha}$. The recovery of $\widehat{\Delta}$ is affected more by noise, but acceptable accuracy is still achieved. As expected, the rate of convergence decreases as the level of noise increases. In the noise free case, an average of 3.63 iterations were required for the algorithm to converge to sub-frame accuracy.

The accuracy of this algorithm relies heavily on the accurate estimation of the fundamental matrix. We examined the effect of estimating the fundamental matrix from stationary points affected by noise and the importance of the position of the selected stationary points relative to the trajectory. In the following experiments, twelve points (the corners of the football goals and line markings shown in Fig. 2) were used to estimate the fundamental matrix, with each point perturbed by isotropic Gaussian noise of various σ values. We conducted three sets of experiments: first with the goals the same as in the previous sub-section; second, with the goals one-tenth of the size; third, with the goals distant from the trajectory. Table 2 shows the mean error in estimating the ratio of frame rates and frame offset over 100 trials. Again, we used $\bar{\alpha} = 5/6$ and $\bar{\Delta} = 3.5$, and no noise was added to the points in the trajectories.

Table 1. Results for recovering $\widehat{\alpha}$ and $\widehat{\Delta}$ from m pairs of corresponding frames when isotropic Gaussian noise of standard deviation σ is added to the trajectory, with the number of iterations required for convergence shown. The errors shown are the mean over 100 trials.

σ	m	$\widehat{\alpha}$ error (%)	$\widehat{\Delta}$ error (frames)	Iterations
0.25	2	0.261	0.144	3.8
0.25	5	0.059	0.032	3.8
0.25	8	0.042	0.021	3.8
0.50	2	0.415	0.236	4.0
0.50	5	0.174	0.089	4.0
0.50	8	0.093	0.047	4.0
1.00	2	1.624	0.802	4.2
1.00	5	0.346	0.192	4.2
1.00	8	0.226	0.146	4.2

Table 2. Mean errors in estimating $\widehat{\alpha}$ and $\widehat{\Delta}$ when the fundamental matrix is estimated from points affected by noise. The synchronization parameters were calculated from 5 corresponding frames, over 100 trials.

Goals	σ	$\widehat{\alpha}$ error (%)	$\widehat{\Delta}$ error (frames)	Iterations
Standard	0.25	0.430	0.189	3.84
Standard	0.50	0.861	0.366	3.89
Standard	1.00	1.703	0.690	3.65
Small	0.01	2.789	1.285	3.40
Small	0.05	7.873	3.121	3.79
Distant	0.25	1.168	0.538	3.79
Distant	0.50	1.688	0.702	3.77
Distant	1.00	3.512	1.621	3.63

From the results in Table 2, it is clear that suitable stationary points are required for estimating the fundamental matrix. This is highlighted in the Small and Distant cases. In the former case, the image of the goals spanned only ten pixels vertically, and less than one hundred horizontally, so the level of noise was significant. In the Distant case, the goals were placed 100m away from the trajectory in the virtual world, so the points used to estimate $\mathtt{F}$ were not close to the trajectory. Hence, it is essential that the stationary points used for estimating the fundamental matrix are not only spatially well separated, but also surround the trajectory in each dimension in order to achieve accurate synchronization. It can also be noted that fewer iterations were required on average for convergence than in Table 1; in this case, the trajectory was not perturbed by noise, leading to a more accurate estimation of the ball's velocity, and faster convergence.

3.2 Real Video Sequences

The algorithm was tested on real video sequences recorded on videos filming at 25 frames per second. Each interlaced frame was separated into two independently recorded fields, so our data was captured at 50 fields per second. A number of stationary back-

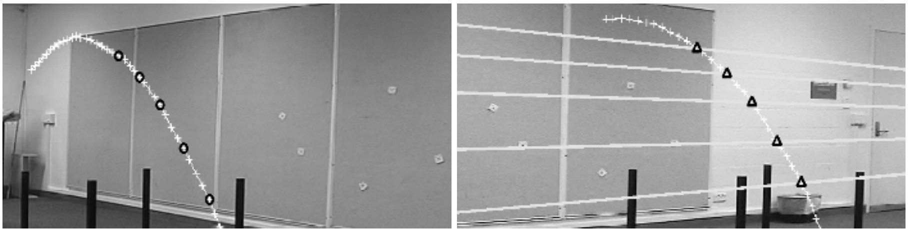


Fig. 3. Two views of the *indoor* sequence. The five corresponding frames used for synchronization are indicated by circles in the left image, with the corresponding epipolar lines shown in the right image, and the ball's position indicated by triangles.

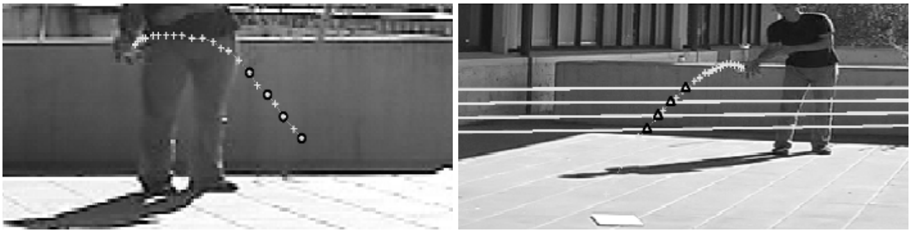


Fig. 4. Two views of the *outdoor* sequence, zoomed in for greater detail. The image markings are the same as in Fig. 3.

ground points were manually selected, and a ball-tracking algorithm [18] was used to track the centroid of the ball through the video sequences.

Figs. 3 and 4 show frames taken from video sequences, with the ball's trajectory overlaid. The ground truth synchronization was obtained by manually locating frames in which the ball bounced. We display the data used for synchronization; in one view, points on the trajectory are shown, and in the other view, the epipolar lines corresponding to these points are shown, and also the point where the lines intersect with the trajectory. Table 3 summarizes results of synchronizing the real video sequences. Where

Table 3. Results of synchronizing two real video sequences, compared with manual synchronization based on frames where the ball bounced. Where $\bar{\Delta}$ is given as a range, accurate manual synchronization was not possible because the ball bounced between frames.

Scene	$\bar{\alpha}$	$\widehat{\alpha}$	$\bar{\Delta}$ (frames)	$\widehat{\Delta}$ (frames)
Outdoors	1.00	0.97	4-5	4.26
Outdoors	1.00	Forced 1	4-5	4.52
Outdoors	2.00	1.94	5-6	5.16
Indoors	1.00	1.04	-8	-7.64
Indoors	1.00	Forced 1	-8	-8.33
Indoors	2.00	2.06	-7	-6.73

$\bar{\alpha}$ is given as 2, we discarded even-numbered frames from one sequence to test the estimation of $\widehat{\alpha}$. "Forced 1" indicates that we assumed that $\bar{\alpha} = 1$, and estimated $\widehat{\Delta}$ accordingly. This knowledge makes the estimation of $\widehat{\Delta}$ more accurate.

4 Discussion

Our algorithm assumes that the ball moves along a single ballistic trajectory (with one upward motion segment and one downward segment) in a short video sequence. This ensures that the epipolar line intersects the trajectory only once in each segment. If a video contains multiple such trajectories, then an ambiguity may arise if the trajectory intersects the epipolar line multiple times. When an epipolar line does cross the trajectory twice, we must determine which intersection is correct. Since we assume that the cameras have the same vertical orientation, we can impose a constraint that the direction of the ball's vertical motion in $\mathcal{S}'_j$ matches that in $\mathcal{S}_i$. It should be noted that any algorithms based on epipolar geometry and using reprojection error as a measure of synchronization are bound to have an ambiguity when an epipolar line crosses a trajectory multiple times. This is also observed by Carceroni et al. [10]. We note that the epipolar constraint is a necessary, but not sufficient, condition for synchronization.

Care must be taken when choosing the frame $\mathcal{S}_i$ for synchronization. There may be a point on the trajectory such that the ball's approximated trajectory $\mathbf{t}'$ in $\mathcal{S}'_j$ is parallel to the epipolar line $\mathbf{l}'_i$ corresponding to the ball's location in $\mathcal{S}_i$. In this case, Step 3 in Section 2.1 yields a point $\mathbf{p}'$ at infinity, and there is no finite solution for the number of frames until the ball crosses the epipolar line. To resolve this problem, a different frame $\mathcal{S}_j$ should be chosen such that the ball has significant vertical velocity and is not moving parallel to $\mathbf{l}'_i$. We assume that the vertical separation of cameras is small relative to the lateral separation, hence when the ball has significant vertical velocity, it should not be moving parallel to an epipolar line which is likely to be close to horizontal. Sometimes, an epipolar line may not intersect the trajectory due to errors in estimating $\mathtt{F}$. Again, $\mathcal{S}_i$ should be chosen such that the ball has significant vertical velocity, such that it will not be at the peak of the trajectory, and $\mathbf{l}'_i$ will not be tangential to the trajectory.

5 Conclusions and Future Work

We have presented an algorithm that uses object motion to recover the ratio of frame rates and temporal offset of two video sequences recorded by two stationary cameras with fixed intrinsic parameters. Experiments on synthetic and real data have shown that the algorithm produces promising results. As expected, as the level of noise increases, the algorithm's accuracy decreases gradually, and gracefully. The fundamental matrix plays an important role in our algorithm and needs to be estimated accurately.

Future work will focus on extending the algorithm to synchronize three sequences, and adopting the trifocal tensor in place of the fundamental matrix. We are also interested in how the algorithm can be modified to allow for camera motion; whilst it is clear that a fundamental matrix is required for each pair of frames, it is less clear how to transfer the ball's velocity between frames. We are also working on an alternative algorithm based on one object trajectory without requiring stationary background points.

References

1. Rao, C., Gritai, A., Shah, M., Syeda-Mahmood, T.: View-invariant alignment and matching of video sequences. In: Proceedings of International Conference on Computer Vision. (2003)
2. Kitahara, I., Saito, H., Akimichi, S., Onno, T., Ohta, Y., Kanade, T.: Large-scale virtualized reality. In: Computer Vision and Pattern Recognition Technical Sketches. (2001)
3. Tresadern, P.A., Reid, I.: Synchronizing image sequences of non-rigid objects. In: Proceedings of British Machine Vision Conference. (2003)
4. Caspi, Y., Irani, M.: Spatio-temporal alignment of sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 1409–1424
5. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. 2nd edn. Cambridge University Press (2004)
6. Reid, I.D., Zisserman, A.: Goal-directed video metrology. In: Proceedings of European Conference on Computer Vision, LNCS 1065, Springer (1996) 647–658
7. Stein, G.P.: Tracking from multiple view points: Self-calibration of space and time. In: Proceedings of Computer Vision and Pattern Recognition. (1999) 521–527
8. Lee, L., Romano, R., Stein, G.: Monitoring activities from multiple video streams: Establishing a common coordinate frame. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 758–767
9. Pooley, D.W., Brooks, M.J., van den Hengel, A.J., Chojnacki, W.: A voting scheme for estimating the synchrony of moving-camera videos. In: Proceedings of ICIP. (2003)
10. Carceroni, R.L., Pádua, F.L.C., Santos, G.A.M.R., Kutulakos, K.N.: Linear sequence-to-sequence alignment. In: Proceedings of CVPR. (2004) 1:746–753
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
12. Tuytelaars, T., Van Gool, L.: Synchronizing video sequences. In: Computer Vision and Pattern Recognition. (2004) 762–768
13. Wolf, L., Zomet, A.: Correspondence-free synchronization and reconstruction in a non-rigid scene. In: ECCV Workshop on Vision and Modelling of Dynamic Scenes. (2002)
14. Giese, M.A., Poggio, T.: Synthesis and recognition of biological motion patterns based on linear superposition of prototypical motion sequences. In: Proceedings of the IEEE Workshop on Multi-View Modeling and Analysis of Visual Scene. (1999) 73–80
15. Yan, J., Pollefeys, M.: Video synchronization via space-time interest point distribution. In: Proceedings of Advanced Concepts for Intelligent Vision Systems. (2004)
16. Laptev, I., Lindeberg, T.: Space-time interest points. In: Proceedings of International Conference on Computer Vision. (2003) 1:432–439
17. Sinha, S.N., Pollefeys, M.: Synchronization and calibration of camera networks from silhouettes. In: International Conference on Pattern Recognition. (2004)
18. Wedge, D., Huynh, D., Kovesi, P.: Tracking footballs through clutter in broadcast digital videos. In: Image and Vision Computing New Zealand. (2004) 155–160

Landmark Based Global Self-localization of Mobile Soccer Robots

Abdul Bais[1],* and Robert Sablatnig[2]

[1] Institute of Computer Technology,
Vienna University of Technology,
Vienna, Austria
bais@ict.tuwien.ac.at

[2] Pattern Recognition and Image Processing Group,
Institute of Computer Aided Automation,
Vienna University of Technology,
Vienna, Austria
sab@prip.tuwien.ac.at

Abstract. We present a stereo vision based global self-localization strategy for tiny autonomous mobile robots in a well-known dynamic environment. Global localization is required for an initial startup or when the robot loses track of its pose during navigation. Existing approaches are based on dense range scans, active beacon systems, artificial landmarks, bearing measurements using omni-directional cameras or bearing/range calculation using single frontal cameras, while we propose feature based stereo vision system for range calculation. Location of the robot is estimated using range measurements with respect to distinct landmarks such as color transitions, corners, junctions and line intersections. Unlike methods based on angle measurement, this method requires only two distinct landmarks. Simulation results show that robots can successfully localize themselves whenever two distinct landmarks are observed. As such marked minimization of landmarks for vision based self-localization of robots has been achieved.

1 Introduction

In mobile robotics the basic requirement for autonomous navigation in any environment is self-localization. There are two different approaches for position estimation: global position estimation and local position tracking. Methods for local position tracking suffer from accumulation of minute measurements to obtain the final estimate, whereas, techniques for global position estimation, are less accurate and often require significantly more computational power [1].This leads to techniques [2, 3, 4, 5, 6, 7, 8, 9, 10] where local measurements are fused with measurements from the robot environment. However, the robot must be able to estimate its position from the very beginning or when/if it loses track of its position during navigation.

* Supported by Higher Education Commission of Pakistan.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 842–851, 2006.

Currently, soccer robots of the size of Tinyphoon are marked on their top with special patterns, which are then tracked for position estimation using a global camera and a host computer. We aim at a shift towards complete autonomy, where all sensing and processing is onboard. However, we look at localization techniques of other (much bigger and slower) soccer robots and also of indoor robots.

Gutmann et al. use a localization method based on dense range scans of the surrounding walls [11]. Whereas, in another approach line segments from range data are matched with the field model to estimate robot position [12]. These methods require that the environment must be surrounded by rectangular walls.

There are several approaches using omni-directional cameras. The major advantage of these approaches is that the robot has a panoramic view of its environment and consequently can acquire more landmark features. Marques and Lima [13] detect field lines using the Hough transform [14] and correlate them with the field model to estimate the robot position. In [8] odometry is used to calculate the expected position of landmarks and then a local search algorithm finds their exact position. Whereas, Motomura et al. localize their robots using dead-reckoning and angle measurements between two landmarks [5].

Approaches using single frontal cameras in conjunction with odometric sensors are widely used for self-localization. These methods are either based on calculating range and bearing based on known shape and size of landmarks or enforce special constrains on environment features[15, 7, 16, 17].

Herrero-Pérez et al. [18] detect features such as goal posts and corners made by the field lines. These features are treated as landmarks in a technique that uses fuzzy logic to account for errors and imprecision in visual recognition.

Approaches using omni-directional cameras with viewing angle of 360 °provide more landmarks but suffer from high cost of the mirror, low resolution of the camera, and requirement of an additional space to fit the mirror and the camera. With frontal cameras one can have high resolution but the field of view is limited. Furthermore, range measurement using single image is too erroneous and the approach cannot be used all the time [19].

To overcome these limitations we propose a stereo vision system with pivoted camera head. This approach would enable us to measure the distance to landmarks and to use bi/trilateration approach to calculate robot position. The pivoted camera head enables the robot to have a 360 °view of its environment. The major advantage of our approach is that it requires less landmarks as compared to the angle based methods.

In this paper we focus on global localization using stereo range measurements. The robot environment consists of visual landmarks i.e. lines, corners, junctions, line intersections and color transitions [20]. The test bed for our algorithm is a soccer playing robot called Tinyphoon (`http://www.tinyphoon.com`) [21].

The balance of the paper is organized as follows: Section 2 discusses robot localization using range or bearing to distinct features in the environment. Potential landmarks are discussed in Section 3. Experimental results are presented in Section 4, finally the paper is concluded in Section 5.

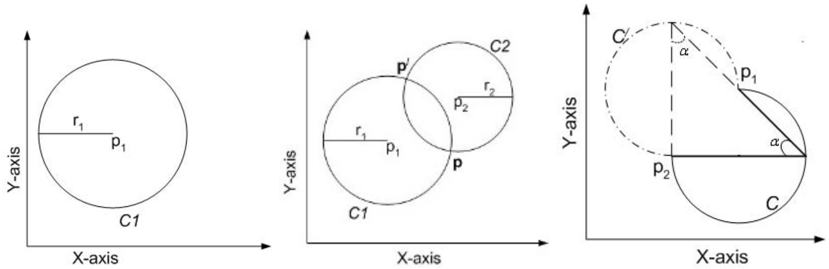
2 Landmark-Based Methods

Landmarks are distinct features that a robot can recognize from its sensory input. Landmarks can be geometric shapes (e.g., rectangles, lines, circles), and they may include additional information (e.g., in the form of bar-codes). In general, landmarks have a fixed and known position, relative to which a robot can localize itself [1].

The input data for position estimation in landmark-based systems may be of range or bearing type. This leads to two different techniques, trilateration and triangulation, respectively. Trilateration is the determination of a robot's position based on distance measurements to known landmarks, whereas, in triangulation, bearing to different landmarks in the environment is used [1].

Fig 1(a) shows the case when the robot identifies a landmark, p_1, and measures the distance r_1. This constrains the robot position to a circle, *C1*. Similarly, detection of landmark point p_2 and its measured distance r_2 will constrain the position of the robot to a circle *C2*. If two points, say p_1 and p_2, are detected at one time then the robot position will be constrained to two points $\mathbf{p}$ or $\mathbf{p}'$, determined by the intersection of the circles *C1* and *C2* (see Fig 1(b)). The ambiguity between these two points can be resolved by considering a fixed order of landmarks.

Fig 1(c) illustrates the case when the robot can only measure the angle α between two landmarks p_1 and p_2. The angle between p_1 and p_2 remains equal to α if the robot is moving along the circular arc *C* or *C'* (shown dotted in Fig 1(c)) [22, 23]. In this case there are infinite number of possible positions and the robot must detect a third landmark point.



(a) The robot is somewhere on the circle

(b) Robot position constrained to two points

(c) The robot position is constrained to circular arcs

Fig. 1. Constraining robot position with landmarks

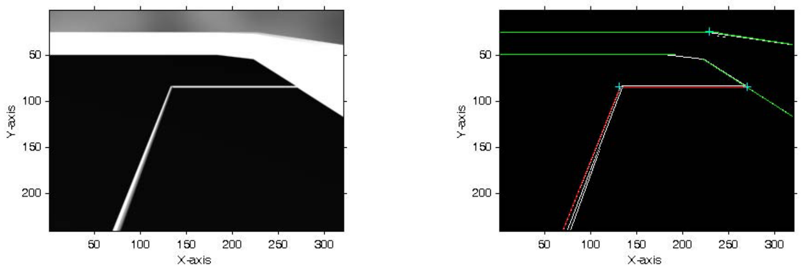
Error free measurements will result in perfect localization. However, measurements are never perfect and errors in distance and angle estimates can vary significantly [23]. Position of the robot will be constrained to a thick ring instead of a perfect circle if there is any error in distance measurement. The intersection of such thickened circles/rings will determine the uncertainty in robot position when two or more landmarks are used.

In addition to measurement errors there could be error in landmark identification and matching with the world map. For the identification errors, some landmarks may not be detected, and some spurious landmarks may be detected. Errors in correspondence could be such that what has been identified as point x on the map may really be point y [23].

3 Landmarks for Self-localization

We use color transitions, corners, junctions and line intersections as landmarks. These landmarks are detected using semantic interpretation of line segments extracted using gradient based Hough transform [20]. Fig 2 illustrate detection of these features.

The vertical edges of the goal corners are normally missed during edge detection and consequent line segments extraction as the change in y-channel value between white and yellow is not significant and the length of the edge is small as compared to other lines in the environment. Therefore, we extract goal corners based on color transitions as discussed in the following section.



(a) Left camera image

(b) Detected features superimposed over the edge map

Fig. 2. Line based landmarks for self-localization

3.1 Detecting Goal Corners

Goals are marked with different colors (blue and yellow). We use color segmentation of the camera images to detect corners of the goal. The process is outlined as follows.

In the left camera image, pixels are tested if they belong to either blue or yellow color. This 'segmentation' is done at a lower scale. Every fourth pixel in a row of every fourth row is tested, which results in a rectangular window around the blue or yellow color patches, if any. The neighborhood of this 'rectangular' window is searched for color transition (color transition from white to yellow, yellow to white, white to blue, or blue to white represents a goal corner), using a full scale. If a color transition is detected in the left image, the corresponding feature points are searched in the right image. The search in the right image is based on parameters of the feature points in the left image. If the corresponding feature point is detected in the right image, its distance from the current robot position is calculated. Detection of two such points determine the robot position as shown in Fig 1(b).

The use of two colors to detect a transition makes the process robust against outliers. All rows inside the rectangular window are searched for transition pixels. One value in a group of pixels is taken as the x-component of the edge between the wall and the colored goal. Outliers in the group are eliminated using simple statistical measures. The calculated stereo range is used to estimate robot position and orientation as discussed in the following sections.

3.2 Calculating Robot Position

We assume that the robot's motion is two dimensional where pose of the robot has 3 degrees of freedom i.e. x, y and θ. The global coordinate system is represented by X and Y axis, whereas the robot coordinate system by x_r and y_r axis. Rotation of robot coordinate system with respect to the global coordinate system is represented by the angle θ. Suppose the robot detects two distinct landmark points p_1 and p_2 at (x_1, y_1) and (x_2, y_2) in the global coordinate system and measures their distances r_1 and r_2, respectively. The two circles at p_1 and p_2 can be described by (1) and (2) as follows.

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2 \tag{1}$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2 \tag{2}$$

Solution of these equations, which is the intersection of the two circles, will give the possible robot position in the global coordinate system. Subtracting (2) from (1) and re-arranging terms we have

$$x = A + By \tag{3}$$

where

$$A = \frac{r_1^2 - r_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2}{2(x_2 - x_1)}$$

$$B = \frac{y_1 - y_2}{x_2 - x_1}$$

further simplification results in

$$y = \frac{-D \pm \sqrt{D^2 - 4CE}}{2C} \tag{4}$$

where

$$\begin{aligned} C &= B^2 + 1 \\ D &= 2AB - 2x_1B - 2y_1 \\ E &= A^2 + x_1^2 - 2x_1A - r_1^2 \end{aligned}$$

One of the solution pairs (p_{x1}, p_{y1}) and (p_{x2}, p_{y2}) (if any) from (3) and (4) will qualify for the possible robot position. The ambiguity between the two positions is resolved by considering a fixed order of landmark points.

3.3 Calculating Robot Orientation

In this section we discuss calculation of robot orientation with respect to goal corners which is done after position is estimated. The process is illustrated in Fig 4. When robot calculates its position with respect to the blue goal θ can be calculated using (5) or (6). One of these equations is used depending on the y coordinate of the robot position.

$$\theta = \alpha_1 - \alpha_2 \tag{5}$$

$$\theta = -(\alpha_1 - \alpha_2) \tag{6}$$

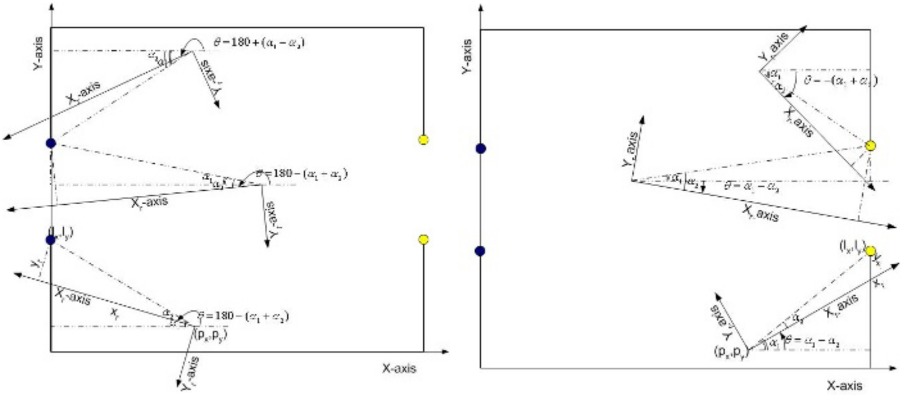
where $\alpha_1 = \arctan(\frac{l_y - p_y}{l_x - p_x})$ and $\alpha_2 = \arctan(\frac{y_r}{x_r})$.

In these equations (l_x, l_y) is the location of one of the landmarks, (p_x, p_y) is robot position and (x_r, y_r) is the location of the selected landmark in robot coordinate system. The landmark and robot position are in global coordinate system. Similarly when the robot position is calculated with respect to the yellow goal θ can be calculated using (7) or (8) as shown in Fig 3(b).

$$\theta = 180\,^\circ - (\alpha_1 + \alpha_2) \tag{7}$$

$$\theta = 180\,^\circ + (\alpha_1 - \alpha_2) \tag{8}$$

where $\alpha_1 = \arctan(\frac{l_y - p_y}{p_x})$ and $\alpha_2 = \arctan(\frac{y_r}{x_r})$.



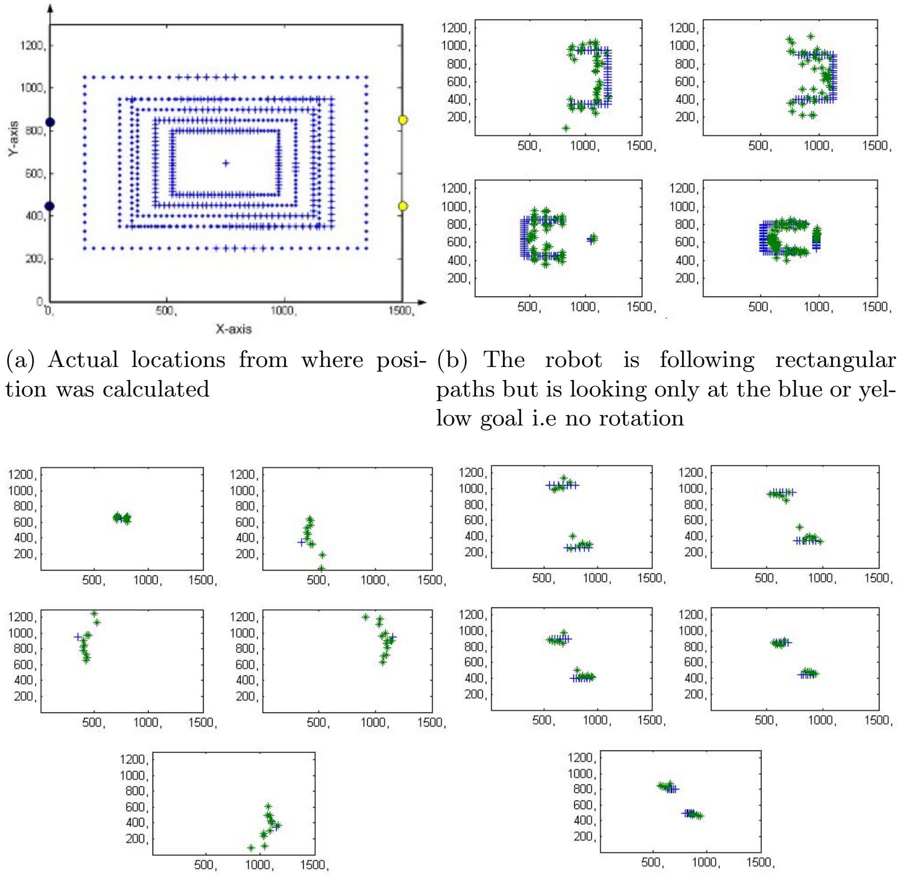
(a) Blue goal (b) Yellow goal

Fig. 3. Robot orientation with respect to goal corners

4 Experimental Results

We simulate the performance of our algorithm using only goal corners as landmarks. We have conducted 14 trials where each one has 100 steps. These trials are further grouped into motion without rotation, rotation without motion and motion with rotation. At every step the robot is taking images of its environment, search for color transitions and calculates its position if it finds both the corners.

Fig 4(a) shows the path followed by the robot. Locations where images were taken and searched for color transitions are shown as dots $(\cdot)$. However, depending on the instantaneous pose of the robot both corners are not visible all the time therefore robot pose is estimated only at limited locations shown as plus $(+)$ superimposed on the dots$(\cdot)$.



(a) Actual locations from where position was calculated

(b) The robot is following rectangular paths but is looking only at the blue or yellow goal i.e no rotation

(c) In this case the robot is following rectangular paths and is rotating as well

(d) The robot is rotating in small steps but without any motion

Fig. 4. Experimental trials

Table 1. Error in x,y and θ for the motion only case

	Mean	Std	Min	Max
δx	63.55	37.10	0.46	157.5
δy	58.99	49.42	0.01	247.14
$\delta\theta$	3.19 °	2.19 °	0.04 °	11.50 °

Table 2. Error in x,y and θ for the rotation only case

	Mean	Std	Min	Max
δx	58.00	26.90	8.10	123.91
δy	78.53	73.52	0.51	240.00
$\delta\theta$	2.05 °	1.58 °	0.03 °	8.53 °

Table 3. Error in x,y and θ for the final case (motion and rotation)

	Mean	Std	Min	Max
δx	36.01	14.32	3.48	69.86
δy	35.92	30.98	1.03	166.24
$\delta\theta$	3.09 °	2.52 °	0.03 °	13.40 °

Table 4. Normalized range error

Mean	Std	Min	Max
6.98%	5.26%	0.01%	25.00%

Fig 4(b) shows the case where the robot follows a rectangular path around the field but its orientation remains fixed at 0 °or 180 °. The plus (+) show the actual position whereas the calculated position is shown as star (∗). A rotation-only case is illustrated in Fig 4(c). The robot is placed at five locations: near the four corners and at the center of the field. Both motion and rotation is shown in Fig 4(d). Here in this case the robot is moving on a rectangular path and is rotating in fixed steps. In the motion-only and motion-with-rotation cases the robot follows rectangular paths of different sizes.

Statistical results for error in pose are shown in Table 1, Table 2 and Table 3 for all the three cases as discussed above. Whereas, normalized error in range measurements is shown in Table 4. The first column in all tables show values for the average absolute error. The standard deviation (Std), minimum (Min) and maximum (Max) values for each group are presented in the 2nd, 3rd and 4th columns. Error δx and δy in Table 1, Table 2 and Table 3 is expressed in millimeters.

As can be seen from Table 4 the normalized error of range is very high. This error is due to several reasons, like, we use a narrow baseline stereo since the construction of the robot does not allow the use of a wide baseline. Again, all processing has to be done by the onboard processors we use low resolution images(QVGA, 320 × 240). Moveover, due to the size and concavity of the goal, it is often difficult to determine which point on the goal is being observed. This results in inconsistent ranges and inconsistent landmark positions [15].

5 Conclusion

The method presented in this paper demonstrates that the robot can successfully localize itself with two distinct landmarks. The distinct and bright color of the goals makes them the strongest candidates to be selected as landmarks. Furthermore, calculating robot position and orientation with respect to goal corners is very efficient as only $N/16$ pixels are tested to determine the rectangular boundaries around the color patches (if any), N being the total number of pixels. This results in localization of color patches which are then searched for the actual corners. The error in range estimation is acceptable as we are using just a single shot localization and have not incorporated any kind of temporal redundancy. The robot pose could be refined once a rough estimate is available.

Currently we are working on methods for efficient interpretation of landmarks other than the goal corners, tracking of landmarks, tracking robot position with local sensors and information fusion. Furthermore, we are also investigating self-localization using range measurement to a single landmark where orientation could be obtained with some other means i.e compass or line segments such as the center line of the soccer field.

References

1. Borenstein, J., Everett, H.R., Feng, L.: Navigating Mobile Robots: Systems and Techniques. A. K. Peters, Ltd. (1996)
2. L.Iocchi, Nardi, D.: Hough localization for mobile robots in polygonal environments. Robotics and Autonomous Systems **40** (2002) 43–58
3. Enderle, S., Ritter, M., Fox, D., Sablatnög, S., Kraetzschmar, G., Palm, G.: Soccer robot localization using sporadic visual features. In et al., E.P., ed.: International Conference on Intelligent Autonomous Systems 6 (IAS-6). (2000) 959–966
4. Adorni, G., Cagnoni, S., Enderle, S., Kraetzschmar, G.K.: Vision-based localization for mobile robots. Robotics and Autonomous Systems **36** (2001) 103–119
5. Motomura, A., Matsuoka, T., Hasegawa, T.: Self-localization method using two landmarks and dead reckoning for autonomous mobile soccer robots. In: RoboCup 2003: Robot Soccer World Cup VII. LNCS (2003) 526–533
6. Adorni, G., Cagnoni, S., Mordonini, M.: Landmark-based robot self-localization: a case study for the robocup goal-keeper. In: Proceedings of the International Conference on Information Intelligence and Systems. (1999) 164–171
7. de Jong, F., Caarls, J., Bartelds, R., Jonker, P.: A two-tiered approach to self-localization. In: RoboCup 2001: Robot Soccer World Cup V. LNCS (2002) 405–410
8. Tehrani, A.F., Rojas, R., Moballegh, H.R., Hosseini, I., Amini, P.: Analysis by synthesis, a novel method in mobile robot self-localization. In et al., G.K., ed.: RoboCup 2004: Robot Soccer World Cup VIII. Volume 3276 of LNCS. (2005) 586–593

9. Utz, H., Neubeck, A., Mayer, G., Kraetzschmar, G.: Improving vision-based self-localization. In et al., G.K., ed.: RoboCup-VI. Number 2752 in LNCS (2002) 25–40
10. Christensen, H.I., Kirkeby, N.O., Kristensen, S., Knudsen, L.: Model-driven vision for in-door navigation. Robotics and Autonomous Systems **12** (1994) 199–207
11. Gutmann, J., Schlegel, C.: Amos: Comparison of scan matching approaches for self-localization in indoor environments. In: 1st Euro micro Workshop on Advanced Mobile Robots, IEEE Computer Society Press (1996)
12. Grisetti, G., Iocchi, L., D.Nardi: Global Hough localization for mobile robots in polygonal environments. In: IEEE International Conference on Robotics and Automation (ICRA-02). (2002) 353–358
13. Marques, C.F., Lima, P.U.: A localization method for a soccer robot using a vision-based omni-directional sensor. In et al., P.S., ed.: RoboCup 2000: Robot Soccer World Cup IV. Number 2109 in LNCS (2001) 96–107
14. Duda, R., Hart, P.: Use of the Hough transformation to detect lines and curves in the pictures. Communications of the ACM **15** (1972) 11–15
15. Stroupe, A.W., Sikorski, K., Balch, T.: Constraint-based landmark localization. In Kaminka, G., Lima, P., Rojas, R., eds.: RoboCup 2002:Robot Soccer World Cup IV. Volume 2752 of LNCS., Springer-Verlag (2003) 8–24
16. Bandlow, T., Klupsch, M., Hanek, R., Schmitt, T.: Fast image segmentation, object recognition and localization in a robocup scenario. In: RoboCup-99: Robot Soccer World Cup III. (1999) 174–185
17. Choi, W., Ryu, C., Kim, H.: Navigation of a mobile robot using mono-vision and mono-audition. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC '99). Volume 4. (1999) 686–691
18. Herrero-Pérez, D., Martínez-Barberá, H., Saffiotti, A.: Fuzzy self-localization using natural features in the four-legged league. In et al., D.N., ed.: RoboCup 2004: Robot Soccer World Cup VIII. LNCS, Springer-Verlag (2005) 110 – 121
19. Nickerson, S.B., Jasiobedzki, P., Wilkes, D., Jenkin, M., Milios, E., Tsotsos, J., Jepson, A., Bains, O.N.: The ark project: Autonomous mobile robots for known industrial environments. Robotics and Autonomous Systems **25** (1998) 83–104
20. Bais, A., Sablatnig, R., Novak, G.: Line-based landmark recognition for self-localization of soccer robots. In: IEEE International Conference on Emerging Technologies (ICET '05), Islamabad, Pakistan (2005) 132–137
21. Novak, G., Mahlknecht, S.: TINYPHOON a tiny autonomous mobile robot. In: IEEE International Symposium on Industrial Electronics (ISIE' 05). (2005) 1533–1538
22. Sugihara, K.: Some location problems for robot navigation using a single camera. Computer Vision, Graphics, and Image Processing **42** (1988) 112–129
23. Sutherland, K.T., B.Thompson, W.: Inexact navigation. In: IEEE International Conference on Robotics and Automation (ICRA' 93). (1993) 1–7

Self-calibration Based 3D Information Extraction and Application in Broadcast Soccer Video[*]

Yang Liu[1], Dawei Liang[1], Qingming Huang[2], and Wen Gao[1,2]

[1] School of Computer Science and Technology, Harbin Institute of Technology,
150001 Harbin, China
[2] Graduate School, Chinese Academy of Sciences, 100039 Beijing, China
{yliu, dwliang, qmhuang, wgao}@jdl.ac.cn

Abstract. This paper proposes a new method based on self-calibration to estimate the ball's 3D position in broadcast soccer video. According to the physical limitation, the ball's 3D position is estimated through the camera position and the ball's virtual shadow, which is the point of intersection between the playfield and the line through the camera's optical center and the ball. First, the virtual shadow is computed by the homography between playfield and image plane. For the image having enough corresponding points, the map is determined directly; for those images not having enough these points, their homographies are estimated through global motion estimation. Then, based on self-calibrating for rotating and zooming camera, and the homography, the camera's position in the playfield is estimated. Experiments show that the proposed method can extract ball's 3D position information without referring to other object with assuming height and obtain promising results.

1 Introduction

Soccer is the most popular sports in the world and appeals to plenty of fans. Every year many matches are broadcasted and stored in digital format. In the last decade, to facilitate audience to rapidly access the stream and enjoy the enriched program, researchers have paid attention to these two research fields: in the first field, they aim to provide tools to help users find important events; for the second, they make efforts to extract 3D information from video, in order to analyze the match or let audience better appreciate the match [1-6]. In this paper, we will focus on the second problem. Based on the knowledge of computer vision, the ball's 3D position is estimated, then cartoon is generated for the highlight segment.

According to the data to be processed, current research can be categorized into two classes: The first class focuses on the data shot by special camera; the second class aims at the broadcast video.

For first class, researchers generally use high-speed camera or multiple cameras. Distante use high-speed camera mounted near the base line to detect ball [7,8]. J.Orwell [9,10] and Hideo Saito [11,12] adopt multiple cameras to monitor the match

[*] This work is partly supported by NEC Research China and "Science 100 Plan" of Chinese Academy of Sciences.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 852 – 861, 2006.

and it is relative easier to track ball and players. According to multi view geometry relationship to reconstruct players' and ball's positions [13]. With many restrictions, such as the expensive equipment, researchers focus their research on broadcast video.

Yu and Tong exploit size, color and shape information to detect ball candidates among playfield and find the ball track in image by Kalman filter or particle filter [14, 15, 16]. In order to acquire the ball's real position, Reid uses infinite point light source and the ball's shadow to estimate the ball's 3D position. However, it is difficult to detect the shadow by computer in image. Ohno [5] and Yamada [6] introduce dynamic equation to estimate the ball's position, while it is not easy to acquire the ball's speed. Kim's method can calculate the ball's position, but it needs two channel signals [17]. The most similar work to ours is [18]. The authors utilize similar triangle to estimate ball's position. The method has to face two difficulties. (1) They have to find two objects, which are perpendicular to playfield and have similar view depths. (2) The method has to assume that a player's height is known. To reduce manual interference, in this paper, we propose a new method to estimate the ball's 3D position based on self-calibration, which does depends on much less hypothesis.

The paper is organized as follows. In section 2, the proposed algorithm is introduced in theory. Section 3 presents experimental results. The last section concludes the paper.

2 The Proposed Method

In what follows, we will describe how to utilize visual geometry to estimate the ball's 3D position. First, we derive the computing formula, then the methods of estimating the involved parameters are described respectively.

2.1 The Formula of Estimating the Ball's Position

As it is known, in the soccer broadcast, the main cameras are generally placed on fixed positions, so it is reasonable to assume that most shots are shot by rotating and zooming camera. According to the physical restriction, the ball's 3D position can be estimated from geometry relationship, figure 1 shows the relationship among the objects, including ball, camera position and the plane in which the ball flies. Figure 1a illustrates the case of the ball's height is lower than the camera's position, and figure 1b is the case of the height is higher than the camera's position. It is needed to point out that our proposed method can deal with these two cases. Explanations for figure 1 are as follows.

- Let the playfield plane be the XOY plane of the world reference frame, and the origin is at the center of the base line, and axis X is perpendicular to the baseline.
- The camera is mounted a fixed position with its coordinate $\mathbf{t}_{cw} = (X_c, Y_c, Z_c)^T$ in the world reference frame. Generally, the information is unknown in broadcast video.
- In principle, movement of ball can be categorized two classes: the first is on the ground; the second is the ball flies in the air. In the second case, let $\mathbf{p}_u = (X_u, Y_u, 0)^T$ be the taking-off point and $\mathbf{p}_e = (X_e, X_e, 0)^T$ be the touching-town point. Based on the physical restriction, it is can be assumed that the ball flies

in the plane, which passes the two points and is vertical to the ground. The function of the plane is given by (1)

$$\pi : \begin{cases} k \cdot X + Y + d = 0, k = -(Y_e - Y_u)/(X_e - X_u)\text{, if } X_e - X_u \neq 0 \\ Y = -d \text{ , otherwise} \end{cases} \tag{1}$$

- Virtual shadow is the point $\mathbf{b}_s = (X_s, Y_s, 0)^T$ of intersection between the line, which passes through the camera's position $\mathbf{t}_{cw} = (X_c, Y_c, Z_c)^T$ and the ball's position $\mathbf{b}_w$ in the air, and playfield plane. The line's function is described by (2).

$$l : [X \quad Y \quad Z]^T = [X_c - X_s \quad Y_c - Y_s \quad Z_c]^T \cdot t + [X_s \quad Y_s \quad 0] \tag{2}$$

- Computing the ball's position $\mathbf{b}_w$. Combining function (1) and function (2), we have

$$\mathbf{b}_w = [X \quad Y \quad Z]^T = [X_c - X_s \quad Y_c - Y_s \quad Z_c]^T \cdot t + [X_s \quad Y_s \quad 0]^T$$
$$t = \begin{cases} \dfrac{-d - k \cdot X_s - Y_s}{(Y_c - Y_s) + k \cdot (X_c - X_s)}, \text{ if } X_e - X_u \neq 0 \\ \dfrac{X_s}{X_s - X_c}, \text{ otherwise} \end{cases} \tag{3}$$

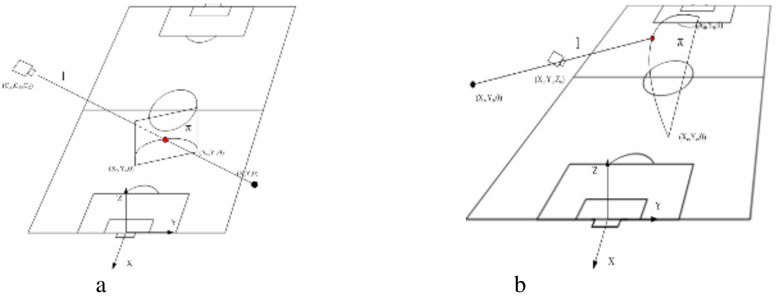


Fig. 1. The geometry relationship for computing the ball's height

From figure 1, we can get the following conclusions: (1) when the ball is on the ground, if we know the transformation between the playfield plane and the image plane, the ball's position can be determined; (2) when ball is in the air, if the positions of camera, the virtual shadow and the plane π are known, then the ball's 3D position can be estimated. The coming subsections specify the calculation of virtual shadow and the estimation of the camera position.

2.2 The Camera Model

Let $\mathbf{M} = (X, Y, Z)$ be a point in space, with the homogenous coordinate $\tilde{\mathbf{M}} = (X, Y, Z, 1)$, and let $\mathbf{m} = (u, v)$ be a point on image, whose homogenous coordinate is $\tilde{\mathbf{m}} = (u, v, 1)$.

According to the pin-hole camera model, the point in space and its image have the following relationship

$$\tilde{\mathbf{m}} \approx \mathbf{K}[\mathbf{R} \quad \mathbf{t}]\tilde{\mathbf{M}}, \tag{4}$$

where ≈ defines two vectors up to a scale factor. In (4), $\mathbf{K}$, called intrinsic matrix, is a 3×3 matrix with the form of

$$\mathbf{K} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

α and β are the horizontal and vertical focal length respectively. (u_0, v_0) is the principal point coordinate. γ is the skewness of the two image axes. In order to use linear method to determine $\mathbf{K}$, assuming that $\gamma = 0$. u_0 and v_0 do not vary with the focal length change. In our system, the principal point is assumed to be at the center of image, and later experiments show that this hypothesis affects the camera position estimation trivially. $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ is the ordinate of the origin of the world reference frame in the camera's reference frame.

2.3 Computing the Virtual Shadow

Without loss generality, the playfield plane can be denoted as $Z = 0$ and is substituted into (4), then we have

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \approx \mathbf{K}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \approx \mathbf{K}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{6}$$

$\mathbf{M}_p = [X, Y]^T$ denotes the point on playfield with its homogenous coordinate $\tilde{\mathbf{M}}_p = [X, Y, 1]^T$, then (6) can be depicted in the concise form

$$\tilde{\mathbf{m}} \approx \mathbf{H}\tilde{\mathbf{M}}_p \quad \text{where} \quad \mathbf{H} \approx \mathbf{K}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]. \tag{7}$$

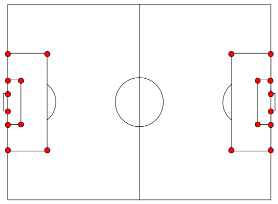


Fig. 2. Soccer playfield model. The red points can be used to calculate an image's homography matrix.

Formula (7) describes the correspondence relationship between two planes. That is to say, when an image's $\mathbf{H}$ is known, for a point in image: if it is a point on the playfield, through (7), its coordinate in world reference frame can be calculated; if the point is not on the playfield, the acquired vector is the coordinate of the virtual shadow. The 3×3 matrix $\mathbf{H}$ is called Homography matrix, which has 8 independent components, so it need 4 pairs of corresponding points to determine $\mathbf{H}$. Figure 2 shows the soccer field model. As [19] regulates that the size of the field is not unique, and only the red points in the figure can be used to compute homography matrix. When an image has enough these points, its $\mathbf{H}$ is computed directly. For the image with insufficient corresponding points, the image's $\mathbf{H}$ is calculated indirectly. As the camera is mounted at a fixed position, the image points $\tilde{\mathbf{m}}_{t-1}$ and $\tilde{\mathbf{m}}_t$ of a still point $\tilde{\mathbf{M}}$ in space in two adjacent frames have the transform

$$\tilde{\mathbf{m}}_t \approx \mathbf{P}_{t,t-1}\tilde{\mathbf{m}}_{t-1}, \tag{8}$$

where $\mathbf{P}_{t,t-1}$ has the similar property with $\mathbf{H}$. In some literatures, $\mathbf{P}_{t,t-1}$ is called inter-frame homography. To differentiate it from $\mathbf{H}$, we call it global motion parameter. Let $\mathbf{H}_{t-1}$ and $\mathbf{H}_t$ are the homography matrixes of frame $t-1$ and frame t respectively. According to (7), we have

$$\begin{cases}\tilde{\mathbf{m}}_{t-1} \approx \mathbf{H}_{t-1}\tilde{\mathbf{M}} \\ \tilde{\mathbf{m}}_t \approx \mathbf{H}_t\tilde{\mathbf{M}}\end{cases}. \tag{9}$$

Substituting (8) into (9), the following recursive function is acquired,

$$\mathbf{H}_t \approx \mathbf{P}_{t-1,t}\mathbf{H}_{t-1} \approx \mathbf{P}_{t-1,t}\mathbf{P}_{t-2,t-1}\mathbf{H}_{t-2} \approx \cdots \approx \mathbf{P}_{t-1,t}\cdots\mathbf{P}_{t-k,t-k+1}\mathbf{P}_{t-k}. \tag{10}$$

Formula (10) tells us that if some image's homgraphy matrix in a video sequence is known, then the $\mathbf{H}$ matrix of image with insufficient corresponding points can be estimated based on (10).

2.4 Camera Position Estimation

Camera position is another important factor for estimating the ball's 3D position. Let us study the relationship between $\mathbf{H}$ in (7) and the intrinsic and extrinsic parameters, then we get

$$\mathbf{K}^{-1}[\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] \approx [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}], \tag{11}$$

in which $\mathbf{h}_i, i = 1,2,3$ is a column of $\mathbf{H}$. According to (11), the camera's extrinsic parameters are calculated by formula (12)

$$\mathbf{t} = s\cdot\mathbf{K}^{-1}\mathbf{h}_3;\ \mathbf{r}_1 = s\cdot\mathbf{K}^{-1}\mathbf{h}_1;\ \mathbf{r}_2 = s\cdot\mathbf{K}^{-1}\mathbf{h}_2. \tag{12}$$

Using the restriction of $\mathbf{R}$ being an orthonormal matrix, where $s = 1/\left\|\mathbf{K}^{-1}\mathbf{h}_i\right\|, i = 1,2$ and $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$ [20]. Here, $\mathbf{t}$ is the coordinate of the origin world reference frame in the camera's reference frame, then the camera's position in the world is

$$\mathbf{t}_{cw} = -\mathbf{R}^{-1}\mathbf{t} \tag{13}$$

It is seen from (12) and (13), in order to compute $\mathbf{t}_{cw}$, an image's $\mathbf{H}$ and the intrinsic parameter $\mathbf{K}$ when capturing the image are known in advance. Since camera exploited in soccer broadcast can think as rotating and zooming camera, we adopt the method proposed in [21]. The principal point is assumed to at the center of image (as literature [22] and our experiments later show that this setting affects the result little), then the components α and β of $\mathbf{K}$ can be acquired by solving equation system (14)

$$\begin{bmatrix} p_{11}p_{21} & p_{12}p_{22} \\ p_{11}p_{31} & p_{12}p_{32} \\ p_{11}p_{31} & p_{22}p_{32} \end{bmatrix} \begin{bmatrix} \alpha^2 \\ \beta^2 \end{bmatrix} = \begin{bmatrix} -p_{13}p_{23} \\ -p_{13}p_{33} \\ -p_{23}p_{33} \end{bmatrix}, \tag{14}$$

where $p_{ij}, i, j = 1,2,3$ is the component of the global motion parameter.

3 Experiments

In this section, we give three experiments. The first experiment is done on the synthesized data and used to verify the proposed method, in which the effect of principal point position is considered. The second experiment is made on real broadcast video, and the goal post height is adopted as estimating object as its height is known. At last the extracted 3D information is applied in highlights cartoon generation.

3.1 Synthesized Data

The synthesized data is generated by a virtual pin-hole camera on a virtual playfield. The playfield is of 100 yard long and 70 yard wide. The world reference frame is set up as figure 1 shows. The virtual camera is mounted at $(-50,-50,10)$, and the initial focal length is $\alpha = 1200, \beta = 1100$. The size of image is 720×572. The camera pans from right to left and the focal length increases 1% per frame. At the same time, random movement is added to tilt angle. The ball flies from (-60,0,0) to (-40,0,0) , the highest point is 8 yard.

First, we consider how the principal point position deviation and noise affect the camera position and the ball's 3D position estimation. The noise is a normal distribution with mean 0 and standard deviation σ . Figure 3a and 3b give the experiment results. At every noise level (11 levels, from 0-2 pixels), 100 runs are done. In the figure, PP denotes the principal point real position, while the principal

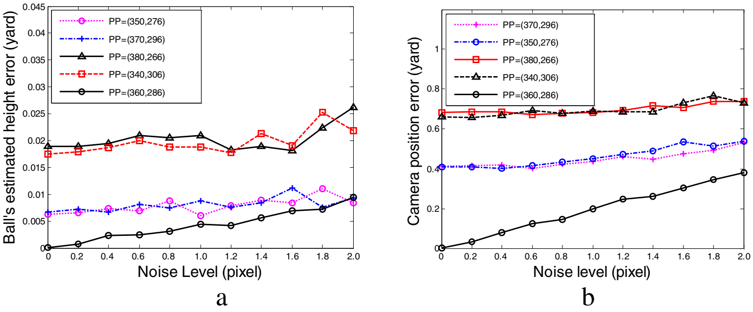


Fig. 3. The effect on camera position (a) and ball position estimation (b) of deviation of principal point position and noise

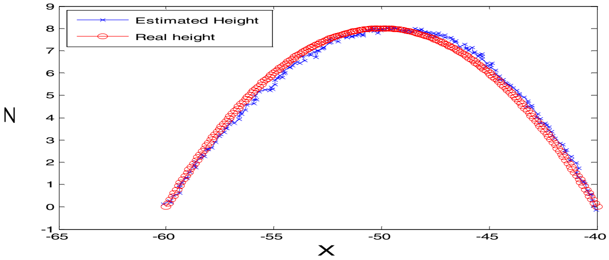


Fig. 4. Ball's height estimation under the condition of its height is higher than the camera position

point is always assumed to be at the centre of image in the calibration process. From the figure, it is concluded that the estimation precision decreases when the deviation of the center from the real PP position increases, and the same to noise level. As the figures show, the effect is trivial (The results are consistent with the report in [22]). So it is reasonable to adopt the center of image as principal point position in practice.

Then the camera is moved to $(-50,-50,5)$, thus highest point of the ball is higher than the camera. We also use the proposed method, under the case of the real PP position is at (340,306) and the noise level is 2 pixels, to estimate the ball's height. Figure 4 gives the result, which indicates that the proposed method still work when the camera is lower than the ball.

3.2 Real Video

In this section, we test the proposed method on real video (352x288). Since these videos are recorded from broadcast, it is impossible to know the ball's real height. We use the method to estimate the goal post height instead of the ball. Figure 5 gives the estimated goal post height, in which the red line indicate the real height (2.44m),the other lines are the estimated goal post height on two sequences (the vertical plane passing through the base line). From the figure, we can find that the estimated value is close to the real value.

At last, we apply the proposed method to extract 3D information from some highlights segment from the last year Europe cup. Figure 6 is a highlight sequence. The black circle is the ball. Through calibration, the camera is at (-51.8,-66.0,22.1), and the unit is yard. The ball in image are detected and tracked by our prior work [23].

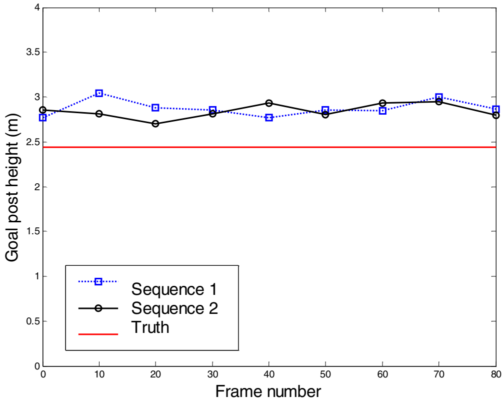


Fig. 5. The estimated goal post height of two sequences

Figure 7 illustrates the ball track in space. Yellow region is the goal area. The green cure is ball track on ground. Red cure is the ball's position in the air after the first pass. Blue cure is ball's position after the goal when the ball is in the air. The six figures depict the scenes from different view points. The ball's flying plane is determined manually.

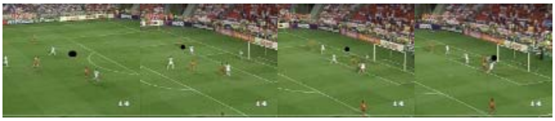


Fig. 6. A highlight sequence. The black disc is the enlarged ball.

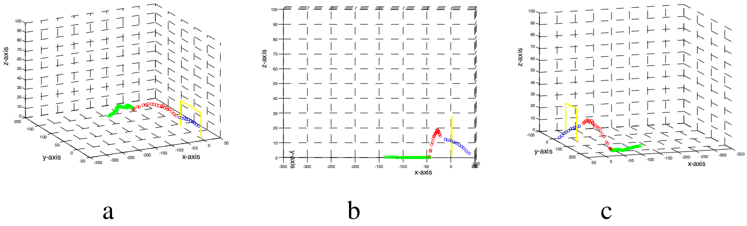


Fig. 7. The ball track from different view point. a: From the main camera. b: Along the base line. c: Opposite side of the camera.

3.3 Cartoon Generation

The extracted ball 3D position and the player's position on the ground are used to generate high light cartoon. Our system allows users to watch the game at any point of view using a 3D viewer based on OpengGL[1].

4 Conclusion

In this paper, we propose a new method to estimate the ball's 3D position from broadcast video based on self-calibration. This method reduces manual interference and does not depend on other object with known height. Experiment show that the method is right in theory, and it can be applied in practical video. At last, the extracted information is used in cartoon generation. This makes audience appreciate match different view point.

References

1. X. Yu, T. S. Hay and H. W. Leong, "3D Reconstruction and enrichment of broadcast soccer video," in Proc. ACM Multimedia, October 2004, New York, NY, USA.
2. T. Bebie and H. Bieri, "Reconstructing soccer game from video sequence," in Proc. of ICIP'1998, pp. 898-902, 1998.2.
3. T. Bebie and H. Bieri, "A Video-Based 3D-econstruction of Soccer games," in EuroGraphics 2000.
4. I. Reid, and A. North, "3D trajectories from a single viewpoint using shadows," British Machine Vision Conference, pp. 863-872, 1998.
5. Y. Ohno, J. Miura and Y. Shirai, "Tracking players and estimation of the 3D position of a ball in soccer games," The International Conference on Pattern Recognition, pp. 145-148, 2000.
6. A. Yamada, Y. Shirai, and J. Miura, "Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games," The International Conference on Pattern Recognition, pp. 303-306, 2002.
7. N. Ancona, G. Cicirelli, E. Stella and A. Ditante, "Ball detection in static images with support vector machines for classification," Image and Vision Computing 21 (2003) 675-692.
8. T. D'Orazio, C. Guaragnella, M. Leo, A. Distante, "A new algorithm for ball recognition using circle Hough transform and neural classifier," Pattern Recognition 37 (2004) 393-408.
9. C. Ren, J. Orwell, G. A. Jones, M. Xu, "A general framework for 3D soccer ball estimation and tracking," In proc. IEEE International Conference on Image Processing 2004.
10. M. Xu, J. Orwell, G. Jones, "Tracking football players with multiple cameras," In proc. IEEE International Conference on Image Processing 2004.
11. S. Iwase, H. Saito, "Parallel Tracking of All Soccer Players by Integrating Detected Positions in Multiple View Images," In Proc. International Conference on Pattern Recognition, 2004.
12. H. Saito, N. Inamoto, S. Iwase, "Sports scene analysis and visualization from multiple-view video," In Porc. IEEE International Conference on Multimedia & Expo, 2004.

[1] http://www.jdl.ac.cn/en/project/mrhomepage/En_demo.htm#video2cartoon.

13. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision* (second edition), Cambridge University Press.
14. X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. W. Wan, “Trajectory-Based Ball Detection and Tracking with Applications to Semantic Analysis of Broadcast Soccer Video,” in Proc. ACM Multimedia03’, November 2003, Berkeley, CA, USA, Pages: 11-20.
15. X. Yu, Q. Tian, and K. W. Wan, “A Novel Ball Detection Framework for Real Soccer Video,” in Proc. ICME’03, vol.2, 6-9 July 2003, pp. 265-268.
16. X. Tong, H. Lu, and Q. Liu, “An Effective and Fast Soccer Ball Detection and Tracking Method,” in Proc. ICPR’04, vol. 4, Aug. 23-26, 2004, Pages: 795-798.
17. H. Kim and K. S. Hong, “Robust image mosaicing of soccer videos using self-calibration and line tracking,” Pattern Analysis & Applications (2001)4:9-19.
18. T. Kim, Y. Seo and K. S. Hong, “Physics-based 3D position analysis of a soccer ball from monocular image sequence,” The International Conference on Computer Vision, pp. 721-726, 1998.
19. FIFA, *Laws of the game*, http://www.fifa.com/en/regulations/regulation/0,1584,3,00.html.
20. Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on Computer Vision, Volume 1, 20-27 Sept. 1999 Page(s):666 - 673 vol.1
21. Y. Seo and K. S. Hong, “Auto-calibration of a rotating and zooming camera,” Proceedings of the IAPR workshop on Machine vision applications, 1998, pp. 274-277.
22. L. Agapito, E. Hayman and I. Reid, “Self-Calibration of Rotating and Zooming Cameras,” International Journal of Computer Vision, 45(2), 107-127, 2001.
23. Liang, D., Liu Y., Huang Q. and Gao W, “A Scheme for Ball Detection and Tracking in Broadcast Soccer Video,” Pacific-Rim Conference on Multimedia, 2005, accepted.

Error Analysis of SFM Under Weak-Perspective Projection

Loong-Fah Cheong and Shimiao Li

Department of Electrical and Computer Engineering,
National University of Singapore,
10 Kent Ridge Crescent, Singapore 119260
{eleclf, shimiao}@nus.edu.sg

Abstract. Despite the long history of studies on weak-perspective projection and the intuitive notion that this model has better numerical stability compared to the perspective model under appropriate conditions, there lacks a deep understanding about the error characteristics of motion and depth recovery process under this model. In this paper, we present the differential approach to SFM under weak-perspective projection. Based on this approach, error behavior which governs the formation of motion ambiguities under weak-perspective is investigated. Recovery of depth information and its distortion are also discussed. Regions where ordinal depth can be extracted is shown to be following a simple relationship under all types of motion, a different result compared to the case of perspective projection.

1 Introduction

Weak-perspective projection model is a good approximation for a full perspective model when field of view is small and the depth relief of the observed object is small compared to its distance from the camera. In this situation, using weak-perspective model can provide us a simpler and often more robust method to solve the structure from motion (SFM) problem. Many works have been done for motion and structure recovery under such model, including the discrete approach under orthographic and weak-perspective projection [1][2][3][4], the continuous approach under orthographic projection [5] and the factorization approach [6]. The latter work has spurred tremendous interest in SFM under situations where orthographic model can be applied. Another motivation for considering weak-perspective model concerns various novel camera systems, whereby dense arrays of cameras have been proposed, some of these inspired by biological visual systems such as compound eyes [7][8]. While these camera systems may have a panoramic field of view, they are made up of discrete units of small visual sensors, each of which might be modeled by a weak-perspective camera.

However, despite all these amounts of work, little has been known on the error characteristics of the motion and depth recovery process under the model. What are the error behaviors governing the recovery of the observable parameters? And how are these behaviors affected by the motion-scene configuration?

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 862–871, 2006.

Only when we have such theoretical analysis, can we fully understand why the model is stable under certain circumstances and what kind of information can be extracted robustly.

A differential approach to SFM problem under weak-perspective is presented in this paper. We formulate the motion field equations and solve the instantaneous motion parameters via the differential epipolar constraint. Based on this, we investigate how the cost function changes its value when errors in the motion estimates or noise in the image measurement arise. This helps us to understand the stability of the motion recovery process and alerts us to specific motion-scene configurations that might be susceptible to errors under weak-perspective model. We also present a method to recover relative depth from relative motion field. The distortion of the recovered relative depth due to errors in the motion estimates is analyzed. It is shown that depth order information can be recovered robustly within a region whose geometry follows a simple relationship unlike the case of the perspective projection [9].

The paper is organized as follows. In section 2, we describe the weak-perspective model and discuss what and how 3D motion parameters can be recovered. Section 3 presents the error analysis on the motion estimates. Section 4 presents method for recovering relative depth and discusses the distortion of the relative depth arising from errors in the motion estimates. Section 5 discusses the robustness of ordinal depth recovery. Experimental result of motion recovery and ordinal depth recovery is presented Section 6 and the conclusions are drawn in Section 7.

2 Motion Recovery Under Weak-Perspective Projection

2.1 Weak-Perspective Projection

We assume the world and the camera coordinate systems are aligned and the camera is calibrated. Consider N points $\mathbf{P}_1, \cdots, \mathbf{P}_N$, where $\mathbf{P}_i = (X_i, Y_i, Z_i)^{\mathbf{T}}$, $i = 1, \cdots, N$, weak-perspective projection is denoted by

$$\mathbf{p}_i = \frac{f}{\overline{Z}} \begin{pmatrix} X_i \\ Y_i \end{pmatrix} \tag{1}$$

where $\mathbf{p}_i = (x_i, y_i)^{\mathbf{T}}$ is the projection of $\mathbf{P_i}$ on the image plane and $\overline{Z}$ is the mean value of $Z_1, \cdots, Z_N$. This projection can be constructed by an orthographic projection of the points onto $Z = \overline{Z}$ plane followed by a perspective projection onto the image plane. The first projection omits the depth relief of the scene and the second scales the points on the plane by a scaling factor $f/\overline{Z}$.

2.2 Weak-Perspective Motion Field Equations

Suppose the camera is undergoing a rigid motion with translational velocity $\mathbf{T} = (T_x, T_y, T_z)^{\mathbf{T}}$, and rotational velocity $\mathbf{\Omega} = (\omega_x, \omega_y, \omega_z)^{\mathbf{T}}$. The relative motion between the camera and point $\mathbf{P}$ is

$$\mathbf{V} = (V_x, V_y, V_z)^{\mathbf{T}} = -\mathbf{T} - \mathbf{\Omega} \times \mathbf{P} \tag{2}$$

Taking time derivatives of both sides of Eq.(1), we get the 2D motion field $(u,v)^{\mathbf{T}}$ under weak-perspective projection as follows:

$$u = sV_x + \delta x, \quad v = sV_y + \delta y \tag{3}$$

where $s = f/\overline{Z}$ is the scaling factor, $\delta = \frac{1}{s}\frac{ds}{dt} = -\frac{V_{\overline{Z}}}{\overline{Z}}$ is the relative changing rate of the scaling factor. Substituting Eq.(2) into Eq.(3), we have

$$\begin{aligned} u &= -t_x - \omega_y z + \omega_z y + \delta x \\ v &= -t_y + \omega_x z - \omega_z x + \delta y \end{aligned} \tag{4}$$

where $t_x = sT_x, t_y = sT_y, z = sZ$. Eq.(4) tells us that under weak-perspective projection, it is impossible to recover the absolute magnitude of the translational components T_x, T_y, the absolute depth Z and the scaling factor s from the motion field since they are coupled together.

2.3 Epipolar Constraint Equation

Eliminating z from equation Eq.(4), we get the differential form of the weak-perspective epipolar constraint:

$$\omega_x u + \omega_y v + (\omega_y \omega_z - \delta\omega_x)\, x + (-\omega_x \omega_z - \delta\omega_y)\, y + \omega_x t_x + \omega_y t_y = 0 \tag{5}$$

The above equation can be written as

$$au + bv + cx + dy + e = 0 \tag{6}$$

which has similar form as its discrete counterpart in [3]. As pointed out in [10], for an image point $(x_i, y_i)^{\mathbf{T}}$, the epipolar constraint equation forces the end of its 2D motion field vector $(u_i, v_i)^{\mathbf{T}}$ to lie on a constraint line in the motion field vector space.

2.4 Cost Function

Given $N\,(N \geq 4)$ points and their weak-perspective motion fields, a, b, c, d, e can be solved from equation Eq.(6) up to an unknown scale factor. In the face of measurement noise in u, v, x and y, the problem can be solved by minimizing the following cost function

$$J = \frac{1}{a^2 + b^2} \sum_{i=1}^{N} (au_i + bv_i + cx_i + dy_i + e)^2 \tag{7}$$

which is the sum of the squared distances between each measured motion field and its constraint line in motion field vector space. Eq.(7) can be simplified by scaling a, b, c, d, e with a scale factor $k = \sqrt{\omega_x^2 + \omega_y^2}$ and redefining a, b, c, d, e by their scaled value as in Eq.(8) such that $a^2 + b^2 = 1$.

$$\begin{aligned} &a = \cos\alpha, \quad b = \sin\alpha, \quad c = \omega_z \sin\alpha - \delta\cos\alpha, \\ &d = -\omega_z \cos\alpha - \delta \sin\alpha, \quad e = t_x \cos\alpha + t_y \sin\alpha \end{aligned} \tag{8}$$

where $\alpha = tan^{-1}(\omega_y/\omega_x)$. From $\partial J/\partial e = 0$, we have $e = -(a\overline{u} + b\overline{v} + c\overline{x} + d\overline{y})$, where $\overline{u}, \overline{v}, \overline{x}, \overline{y}$ are the centroids of the measured data u_i, v_i, x_i, y_i $(i = 1, \cdots, N)$, respectively. Thus, J can be simplified as

$$J = \sum_{i=1}^{N} (au_i' + bv_i' + cx_i' + dy_i')^2 \tag{9}$$

where notation $'$ denotes value with respect to the centroid of the original data. Minimizing J in Eq.(9) is equivalent to getting the least square solution of Eq.(6) under constraint $a^2 + b^2 = 1$, which can be solved using singular value decomposition technique.

2.5 Solving for Motion Parameters

The following motion parameters can be determined from $a, b, c, d, e\big(a^2 + b^2 = 1\big)$, by using Eq.(8):

$$\begin{gathered} \alpha = \tan^{-1}(b/a), \quad \omega_z = bc - ad \\ \delta = -(ac + bd), \quad t_\alpha = (t_x, t_y) \cdot (\cos\alpha, \sin\alpha)^{\mathrm{T}} = e \end{gathered} \tag{10}$$

In other words, given a, b, c, d, e, we can determine:

1. α, the direction of $(\omega_x, \omega_y)^{\mathbf{T}}$ up to a 180° ambiguity;
2. angular velocity ω_z (the cyclotorsion);
3. relative changing rate of the scaling factor δ;
4. t_α, the component of the translational velocity (scaled by s) along the $(\omega_x, \omega_y)^{\mathbf{T}}$ direction.

Recall, however, that a, b, c, d, e are only determined up to an overall scale factor. Thus, we cannot determine k, the magnitude of $(\omega_x, \omega_y)^{\mathbf{T}}$. The translational component perpendicular to the direction $(\omega_x, \omega_y)^{\mathbf{T}}$ is also indeterminate.

3 Error Analysis on Motion Estimates

We denote the estimated parameters with the hat symbol ˆ and errors in the estimated parameters with the subscript e (where error of any estimate is defined as $s_e = s - \widehat{s}$). Symbol˜denotes the measured value. Noise in a measured value is denoted with the subscript n (where noise of any measured value is defined as $s_n = s - \widetilde{s}$). Notation $'$ denotes value with respect to the centroid of the original data.

3.1 Analyzing Cost Function in Noise-Free Case

First, we analyze J in the noise-free case, which means $u_n = v_n = 0$ and $x_n = y_n = 0$. We aim to see how J changes with errors in the 3-D motion estimates.

Substituting Eq.(4) into Eq.(9) and expressing a, b, c, d in terms of the 3D motion estimates, we obtain the cost function J in the following form:

$$J = \sum_{i=1}^{N} \left(\sin\left(\widehat{\alpha} - \alpha\right) k z_i' - \omega_{ze} r_i \sin\left(\widehat{\alpha} - \theta_i\right) + \delta_e r_i \cos\left(\widehat{\alpha} - \theta_i\right)\right)^2 \qquad (11)$$

where $r_i = \sqrt{x_i'^2 + y_i'^2}, \theta_i = \tan^{-1}\left(y_i'/x_i'\right)$ and $z_i' = z_i - \overline{z}$, $\overline{z} = \sum\limits_{i=1}^{N} z_i$. Obviously, for $\forall i, (i = 1, \cdots, N)$, the term in the parentheses in Eq.(11) is a sine function of $\widehat{\alpha}$, which can be readily simplified to

$$J = \sum_{i=1}^{N} \left(A_i \sin\left(\widehat{a} - \psi_i\right)\right)^2 \qquad (12)$$

where for compactness of expression, the magnitude A_i and phase ψ_i are customarily expressed in the corresponding complex form:

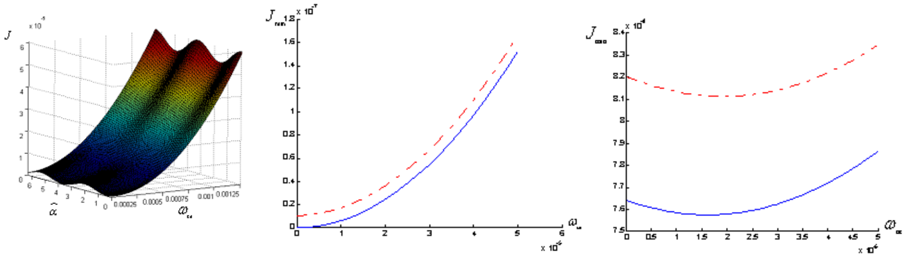
$$A_i \exp\left(j\psi_i\right) = k z_i' \exp\left(j\alpha\right) - \omega_{ze} r_i \exp\left(j\theta_i\right) - \delta_e r_i \exp\left(j\left(\theta_i + \pi/2\right)\right) \qquad (13)$$

$\widehat{\alpha}$ can be taken out the of the summation sign via further manipulation:

$$J = \frac{1}{2}\left(\sum_{i=1}^{N} A_i^2 - \overline{A}^2\right) + \left(\overline{A} \sin\left(\widehat{\alpha} - \Psi\right)\right)^2 \qquad (14)$$

with $\left(\overline{A} exp\left(j\Psi\right)\right)^2 = \sum\limits_{i=1}^{N} \left(A_i \exp\left(j\psi_i\right)\right)^2$. Let $J_{\min} = \frac{1}{2}\left(\sum\limits_{i=1}^{N} A_i^2 - \overline{A}^2\right)$. We have the following observations from the preceding equations:

1. Given fixed values of ω_{ze} and δ_e , the cost function J is a squared sine function of $\widehat{\alpha}$, with minimal value $J_{\min}$ at $\widehat{\alpha} = \Psi$.
2. $J_{\min} = 0$ if and only if all the ψ_i's have the same value. Given uniform feature distribution, this can only happen if and only if $\omega_{ze} = \delta_e = 0$. Under these conditions, $A_i \exp(j\psi_i) = k z_i' \exp(j\alpha)$. J vanishes if and only if $\widehat{\alpha} = \alpha$ (or $\widehat{\alpha} = \alpha + \pi$) and $\omega_{ze} = \delta_e = 0$.
3. Any residual values in ω_{ze} and δ_e will increase the minimum $J_{\min}$ as well as changing the value of Ψ at which this minimum is obtained. Nevertheless, given any fixed ω_{ze} and δ_e, the minima of J are still the minima of the sinusoid (Fig.1(a)), even though the values of these minima monotonically increase as ω_{ze} and δ_e increase (Fig.1 (b)). This should also mean that ω_z and δ can be estimated quite robustly.
4. If the feature points are evenly and densely distributed around their centroid, the shift in Ψ(and hence $\widehat{\alpha}$) caused by nonzero ω_{ze} and δ_e would be significantly ameliorated. On the other hand, local and sparse features would mean that $\widehat{\alpha}$ is more susceptible to errors in ω_z and δ.



(a) (b) (c)

Fig. 1. (a) J as function of $\widehat{\alpha}$ and ω_{ze} ($\delta_e = 0$). (b) In noise free case, J_{min} increases monotonically as ω_{ze} increases, regardless of the value of $\widehat{\alpha}$. (c) Gaussian noise with zero mean and a deviation equal to %10 of the average optical flow is added to the optical flow. The curve is non-monotonic, causing biased estimate in ω_z. (ω_{ze} ranges from 0 to $0.1|\omega_z|$. The solid curve corresponds to $\delta_e = 0$ and the dot-dash curve $\delta_e = 0.1|\delta|$.)

3.2 Role of Noise

In practice, optical flow is always estimated with noise. We express the measured flow which is corrupted by noise as $(u'_{in}, v'_{in})^{\mathbf{T}}$ Recall that the notation $'$ denotes value with respect to the centroid of the original data (see Eq.(9)). It can be proven that the forms of Eq.(12) and Eq.(14) still hold with Eq.(13) changed to

$$\begin{aligned} A_i \exp(j\psi_i) &= kz'_i \exp(j\alpha) - \omega_{ze} r_i \exp(j\theta_i) \\ &- \delta_e r_i \exp(j(\theta_i + \pi/2)) + \|\mathbf{u}'_{in}\| \exp(j(\eta_i + \pi/2)) \end{aligned} \tag{15}$$

where $\|\mathbf{u}'_{in}\| = \sqrt{u'^2_{in} + v'^2_{in}}, \eta_i = \tan^{-1}(u'_{in}/v'_{in})$.

Generally, when noise exists ($\|\mathbf{u}'_{in}\| \neq 0$), ψ_i's cannot be the same for all points. The term $\|\mathbf{u}'_{in}\| \exp(j(\eta_i + \pi/2))$ will pull $A_i \exp(j\psi_i)$ in the $(\eta_i + \pi/2)$ direction resulting in non-zero J_{min}, and a biased estimate of $\widehat{\alpha}$. The minimum of J_{min} might be achieved at certain value of ω_{ze} and δ_e other than $\omega_{ze} = \delta_e = 0$ (Fig.1 (c)).

4 Recovery of Structure

4.1 Recovery of Relative Depth Information

If the camera is undergoing a rigid motion with translation $\mathbf{T}$ and rotation $\mathbf{\Omega}$, the relative motion field between the two image points $\mathbf{p_i}$ and $\mathbf{p_0}$ under weak-perspective projection are

$$\triangle u_i = -\omega_y \triangle z_i + \omega_z \triangle y_i + \delta \triangle x_i, \quad \triangle v_i = \omega_x \triangle z_i - \omega_z \triangle x_i + \delta \triangle y_i \tag{16}$$

where $\triangle u_i = u_i - u_0, \triangle v_i = v_i - v_0, \triangle z_i = z_i - z_0, \triangle x_i = x_i - x_0, \triangle y_i = y_i - y_0$. Thus the relative motion field between image points does not contain the translational part of the motion field. Furthermore, since $\triangle z_i$ is coupled with

ω_x and ω_y,the magnitude $k = \sqrt{\omega_x^2 + \omega_y^2}$ is indeterminate from the motion field. However we can recover the quantity $k\triangle z_i$ under the condition $k \neq 0$ as follows:

$$\begin{aligned} &k\triangle z_i = (ks)\triangle Z_i = (-\triangle u_i \sin\alpha + \triangle v_i \cos\alpha) \\ &+\omega_z(\triangle y_i \sin\alpha + \triangle x_i \cos\alpha) + \delta\,(\triangle x_i \sin\alpha - \triangle y_i \cos\alpha) \end{aligned} \tag{17}$$

If $k = 0$, no depth information can be recovered from the motion field. To sum up, relative depth can only be determined up to a scale factor ks (ambiguity in k results in the bas-relief ambiguity in addition to the overall scale ambiguity s) and a reflection about the plane $Z = Z_0$ (mirror ambiguity as α is recovered with a 180° ambiguity).

We can now recover the relative position of point $\mathbf{P_i}$ with respect to $\mathbf{P_0}$ as $(\triangle x_i, \triangle y_i, k\triangle z_i)$. In this way, the object structure is recovered up to an affine transformation which scales the X, Y, Z axis by the unknown scaling factors $s, s, \pm(sk)$, respectively, where the negative sign in the last scaling factor indicates mirror ambiguity. Given the true position of one point $\mathbf{P}$ with respect to $\mathbf{P_0}$, both s and k can be solved and the Euclidean structure of the object can be recovered. For notational convenience, we use $\triangle z$ to denote $k\triangle z$ in the following discussion.

4.2 Error Analysis on Relative Depth Estimate

If motion parameters are not estimated precisely, error will arise in the estimated relative depth. According to Eq.(17), the estimated relative depth (up to a scale sk) $\triangle\widehat{z}$ between two scene points is

$$\begin{aligned} &\triangle\widehat{z}_i = (ks)\triangle\widehat{Z}_i = (-\triangle u_i \sin\widehat{\alpha} + \triangle v_i \cos\widehat{\alpha}) \\ &+\widehat{\omega}_z(\triangle y_i \sin\widehat{\alpha} + \triangle x_i \cos\widehat{\alpha}) + \widehat{\delta}\,(\triangle x_i \sin\widehat{\alpha} - \triangle y_i \cos\widehat{\alpha}) \end{aligned} \tag{18}$$

Substituting Eq.(16) into Eq.(18) and simplifying, we obtain

$$\triangle\widehat{z} = \triangle z \cos\alpha_e - \omega_{ze} r \cos(\theta - \widehat{\alpha}) + \delta_e \sin(\theta - \widehat{\alpha}) \tag{19}$$

where $r(\cos\theta, \sin\theta) = (\triangle x, \triangle y)$. Eq.(19) can also be written as $\triangle\widehat{z} = \triangle z \cdot D$, where D is the distortion factor given by

$$D = \cos\alpha_e - \frac{r}{\triangle z}\left(\omega_{ze}\cos(\widehat{\alpha} - \theta) + \delta_e \sin(\widehat{\alpha} - \theta)\right) \tag{20}$$

The above tells us how errors in the estimates of motion parameters may distort the recovered relative depth by the multiplicative factor D. For instance, in the case of the 180° ambiguity in the estimate of α, $\alpha_e = \pi$ and $\omega_e = \delta_e = 0$, we have $D = -1$, $\triangle\widehat{z} = -\triangle z$. This case is due to the mirror ambiguity. The object structure is recovered as the reflection of the true structure.

5 Ordinal Depth Recovery

For two points in the scene $\mathbf{P_1}$ and $\mathbf{P_2}$, if and only if the relative depth between $\triangle z$ and the estimate $\triangle\widehat{z}$ have the same sign, we say that the relative ordinal depth between $\mathbf{P_1}$ and $\mathbf{P_2}$ is preserved in the recovered object structure. For $\triangle\widehat{z}$ and $\triangle z$ to have the same sign, the distortion factor D must be positive. If $\cos\alpha_e > 0$, the condition:

$$r/\left|\triangle z\right| < \left|\cos\alpha_e/\sqrt{\omega_{ze}^2+\delta_e^2}\right| \tag{21}$$

ensures that D is positive. On the other hand, if $\cos\alpha_e < 0$, the same condition ensures D is negative. In the latter case, the relative depth order between any two points which satisfy equation Eq.(21) is reversed. Therefore, for ordinal depth to be fully preserved or fully reversed, Eq.(21) must be satisfied for any two points in the scene.

We define the visual angle subtended by two image points as $\tau = 2\tan^{-1}\frac{r}{2f}$ Eq.(21) can be rewritten as

$$\tau < 2\tan^{-1}\left(\frac{\varepsilon k}{2}\left|\frac{\triangle Z}{\overline{Z}}\right|\right) \tag{22}$$

where $\varepsilon = \left|\cos\alpha_e/\sqrt{\omega_{ze}^2+\delta_e^2}\right|$. This tells us that given certain k and ε, if the scaled relative depth $\left|\triangle Z/\overline{Z}\right|$ of two points and their visual angle is such that Eq.(22) is satisfied, then their ordinal depth estimate can be recovered. In other words, given a region subtending an angle τ_i, we can determine the depth order of any two points within this region up to a resolution of $\left|\triangle Z/\overline{Z}\right| = \frac{2\tan(\tau_i/2)}{\varepsilon k}$. The smaller the scaled relative depth we want to resolve, the smaller the region becomes. This agrees with the intuition that in human vision, the depth order of two objects close together can be determined with much greater ease than that of objects far apart. The size of this region is also dependent upon the magnitude of the in-plane rotation k. Lastly, we note that the geometry of this region depends on the various factors in a simple manner, in the sense that it is independent of the image coordinates. This is a virtue against which depths recovered from perspective projection cannot prevail, because in the latter case, the properties of the recovered depths depend critically on the types of motion being executed and in general on the image location [9].

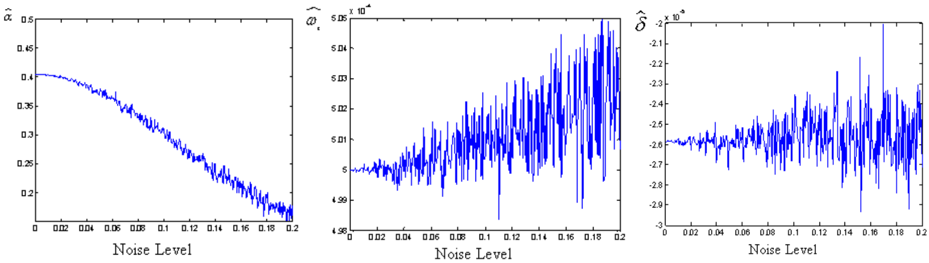
6 Simulation

In this section, we compute motion estimates from synthetic image data and show how the extent of the region within which the ordinal depth can be recovered varies with different errors in the motion estimates and different values of scaled relative depth.

1. 100 points were generated randomly in space within the depth range $[60, 80]$ such that their projections in the image were evenly distributed around the image center.
2. All these points were subjected to a rigid motion with translation $\mathbf{T} = (0.0002, 0.0004, 0.0005)^{\mathbf{T}}$ and rotation $\mathbf{\Omega} = (0.0007, 0.0003, 0.0005)^{\mathbf{T}}$(or $\widehat{\alpha} = 0.4049$, $k = 7.6158\times10^{-4}$, $\delta = -2.5728\times10^{-5}$). 2D motion fields were then generated using weak-perspective projection.

3. Gaussian noise with zero mean was added to each component of the 2D motion field with different noise levels. The standard deviation of the noise varied from 0 to 20% of the average magnitude of the optical flows. For each noise level, the corrupted flows were used to compute the motion parameters by the method described in Section 2.
4. The extent of the region within which the ordinal depth can be recovered was computed using Eq.(22), for different resolutions of $|\triangle Z/\overline{Z}| = \frac{1}{20}, \frac{1}{200}, \frac{1}{2000}$. What these values of $|\triangle Z/\overline{Z}|$ mean is as follows. For instance, if a weak-perspective model is applied to a scene where the maximum depth variation compared to the average depth is about $\frac{1}{20}$, using a value of say $|\triangle Z/\overline{Z}| = \frac{1}{2000}$ in Eq.(22) would mean that we want to further resolve these depths into 100 different depth levels. The results are shown in Fig. 3.

Fig. 2 shows that as noise level increases, all motion estimates shift gradually, with $\hat{\omega}_z$ and $\hat{\delta}$ exhibiting stable recovery (error less than %1 in $\hat{\omega}_z$ and %15 in $\hat{\delta}$ given noise level of %20).



(a) $\hat{\alpha}$ (b) $\hat{\omega}_z$ (c) $\hat{\delta}$

Fig. 2. Result of motion estimates for different noise levels

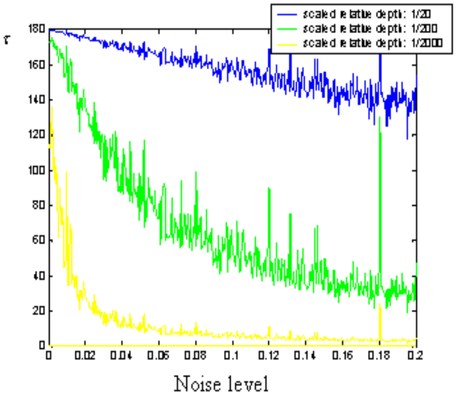


Fig. 3. Extent of the region within which ordinal depth can be recovered, expressed as visual angle subtended by the region. Noise level is expressed as percentage of the average magnitude of the optical flows, ranging from 0 to %20. $|\triangle Z/\overline{Z}|$ =1/20,1/200,1/2000.

As can be seen from Fig. 3, even given a substantial level of noise, the extent of the region within which ordinal depth can be recovered up to a fine degree is still quite sizable. Even at a desired resolution of $\left|\triangle Z/\overline{Z}\right|$ =1/2000 and noise level of %20, the extent of the region is approximately 30°. This is good news for weak perspective model. The jagged nature of the curves stems from the noisy flows affecting the motion estimates in a stochastic manner.

7 Conclusion and Future Work

Based on the proposed differential approach, we investigate the error characteristics of SFM under weak perspective projection. Motion estimates are shown to degenerate in a graceful manner with the presence of noise. The nature of depth distortion under weak perspective model are also elucidated for the first time. It is shown that, for a given level of depth resolution, ordinal depth information can be recovered robustly within an image region satisfying a simple relationship and occupying a significant spatial extent in practice. Future work will be done in adapting the analysis to discrete approach and to images under perspective projection.

References

1. S. Ullman, *The interpretation of visual motion.* M.I.T. Press, Cambridge, MA, 1979.
2. T. S. Huang and C. H. Lee, “Motion and structure from orthographic projections,” *IEEE Trans. PAMI*, 11(5), pp.536-540, 1989.
3. L. S. Shapiro, A. Zisserman, and M. Brady, “3D motion recovery via affine epipolar geometry,” *IJCV*, 16(2), pp. 147-182, 1995.
4. I. Shimshoni, R. Basri and E. Rivlin, “A geometric interpretation of weak-perspective motion,” *IEEE Trans. PAMI*, 21(3), pp. 252-257, 1999.
5. D. D. Hoffman, “Inferring local surface orientation from motion fields,” *J. Opt. Soc.* vol. 72, no. 7, pp. 888-892, 1982.
6. C. Tomasi and T. Kanade, ”Shape and motion from image streams under orthography: a factorization method,” *IJCV*, 9(2), pp. 137-154, 1992.
7. J. C. Yang, M. Everett, C.Buehler and L.McMillan, ”A real-time distributed light field camera,” *Proceedings of the 13th Eurographics workshop on Rendering, Pisa, Italy*pp. 77-86, 2002.
8. J. Neumann, C. Fermuller and Y. Aloimonos, ”Polydioptric Camera Design and 3D Motion Estimation,” *CVPR*, (2), pp. 294-304, 2003.
9. L.F. Cheong and T. Xiang ”Characterizing Depth Distortion under Different Generic Motions,” *IJCV*, 44(3), pp 199-217, 2001.
10. J. Weber and J. Malik, ”Rigid Body Segmentation and Shape Description from Dense Optical Flow under Weak Perspective,” *IEEE Trans. PAMI*, 19(2), pp. 139-143, 1997.

General Specular Surface Triangulation

Thomas Bonfort, Peter Sturm, and Pau Gargallo

MOVI - GRAVIR - INRIA - 38330 Montbonnot, France
http://perception.inrialpes.fr

Abstract. We present a method for the reconstruction of a specular surface, using a single camera viewpoint and the reflection of a planar target placed at two different positions. Contrarily to most specular surface reconstruction algorithms, our method makes no assumption on the regularity or continuity of the specular surface, and outputs a set of 3D points along with corresponding surface normals, all independent from one another. A point on the specular surface can be reconstructed if its corresponding pixel in the image has been matched to its source in both of the target planes. We present original solutions to the problem of dense point matching and planar target pose estimation, along with reconstruction results in real-world scenarii.

1 Introduction

Reconstructing surfaces from images usually relies on the identification and matching of pixels corresponding to a same 3D point on the surface. On unpolished surfaces, matching can be fulfilled by analyzing surface texture, and assuming that identical texture patches correspond to identical points on the surface. In the case of specular surfaces, the apparent surface texture is the reflection of the object's surroundings, being *de facto* viewpoint-dependent, thus invalidating the geometric constraints used by all non-specific reconstruction algorithms. Even standard laser scanners are unable to acquire specular surfaces, as all of the laser energy is reflected symmetrically to the normal of the surface, and therefore cannot be detected by the sensor [1]. Consequently, specularities, and even more importantly specular objects, are usually discarded as noise by most surface reconstruction algorithms. However, specular reflections give rise to strong constraints on surface depth and orientation, and we take advantage of these additional cues to reconstruct a precise model of the surface.

We describe a method recovering points of a specular surface, independently from one another. We assume an internally calibrated pinhole camera viewing the reflection of a planar target, and a dense matching of the camera pixels with the points on the target. While the camera is rigidly attached to the specular surface, we acquire images of the reflection of the target placed at two different unknown locations. The foundation of our method is closely related to the work on general (*i.e.* non central) cameras, as the reconstruction of the specular surface from the images of a calibrated camera is equivalent to the calibration of a non-central catadioptric system. The output of the algorithm is a collection of 3D points of

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 872–881, 2006.

the specular surface, and the two transformations (rigid displacements) from the camera reference coordinate system to the target plane coordinate systems.

1.1 Previous Work

Though less actively than for lambertian surfaces, the reconstruction of specular surfaces from images has interested researchers in the field of computer vision for the past 20 years. For example, Blake and Brelstaff [2] study the disparity of highlights on a specular surface in a stereoscopic framework. Zisserman *et al.* [3] tracked the motion of specularities obtaining a degree-1 family of curvatures along the tracked path.

In [4], Oren and Nayar study the classification of real and reflected features, and recover the profile of a specular surface by tracking an unknown scene point. The work was extended to complete object models by Zheng and Murata in [5], who reconstruct a rotating specular object by studying the motion of the illumination created by two circular light sources.

Halstead *et al.*, in [6], fit a spline surface to a set of normals, iteratively refining the result. Their method requires an initial seed point on the specular surface, and was applied to the sub-micronic reconstruction of the human cornea. The approach was extended by Tarini *et al.* [7] who integrate around a seed point, and use a global self-coherence measure to estimate the correct depth for the seed point. Under a distant light configuration, Solem *et al.* [8] fit a level-set surface with a variational approach.

Savarese *et al.* detail in [9] the mathematical derivations allowing the recovery of surface parameters up to 3$^{\mathrm{rd}}$ order from one view of a smooth specular object reflecting two intersecting calibrated lines, when scale and orientation can be measured in the images.

Bonfort and Sturm [10] present a space carving approach using surface normals instead of color as a consistency measure.

1.2 Notation

The following notation will be used throughout the article: **bold** letters represent a vector in 3D space, while *italic* letters represent scalars. Matrices are represented by $\mathsf{CAPITAL}$ letters.

2 Approach

Suppose a calibrated pinhole camera located at $\mathbf{O_c} = \mathbf{0}^\mathsf{T}$ observing the reflection in an unknown specular surface of a known 3D feature $\mathbf{Q}$. As the camera is calibrated, recovering the position of the surface at the point $\mathbf{p}$ of reflection is simply the estimation of its depth along the corresponding projection ray. This already constrained scenario is still insufficient in order to obtain a solution to the depth estimation, as for every point $\mathbf{P}$ along the projection ray, we can compute a surface orientation that would produce an identical observation: depth estimation of a point on a specular surface from one image gives rise to a one dimensional solution, function of surface depth and orientation.

Now consider the same setup, except that for a given camera pixel **p**, two 3D point correspondences $\mathbf{Q_1}$ and $\mathbf{Q_2}$ are given. This constraint is sufficient to uniquely determine the depth of the specular surface at **p**, namely as the intersection of the lines formed by the camera's projection center and **p** on the one hand, and $\mathbf{Q_1}$ and $\mathbf{Q_2}$ on the other.

If we consider the (camera + specular surface) system as a general camera, finding two points $\mathbf{Q_1}$ and $\mathbf{Q_2}$ for each **p**, and therefore obtaining a reconstruction of the surface, is equivalent to calibrating this camera, as this is usually done as a one-to-one mapping of image pixels with lines in 3D space. In [11] or [12], such a calibration is achieved by using points on calibration planes: pixels in the image are matched with their 2D correspondent in the target planes, then the only step necessary in order to obtain 3D coordinates of these points is to estimate the pose of the planes in the camera reference coordinate system. Figure 1 summarizes our reconstruction method for 3 point correspondences: reconstructing the specular surface sums down to matching camera pixels with their source in the target planes, then estimating the two transformation matrices T_1 and T_2, that map points from the target reference coordinate system to the camera one.

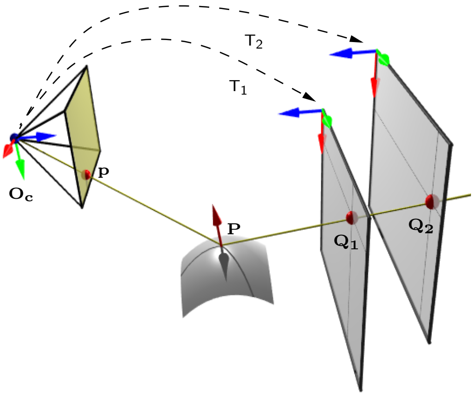


Fig. 1. Reconstruction Approach. Matching of image pixels with their source in the targets and estimating two plane poses is sufficient to reconstruct the surface.

3 Dense Matching

The 3D position of a point on the specular surface corresponding to a given pixel in the camera image plane can only be computed if a correspondence can be found in both of the target planes. As such, in order to obtain a dense reconstruction of the specular surface, each pixel of the specular surface must be matched to its target correspondence.

3.1 Initial Matching

We use a standard computer monitor displaying Gray codes, once original and once inverted [13]. The total number of images taken for each pose of the target is therefore twice the binary resolution in each direction.

The resolution of the codes and the width of the low order stripes must be chosen according to the shape of the specular object and the resolution of the camera. Too high resolution codes tend to be blurred out and become unusable, whereas too coarse ones lack in precision. In most cases, multiple pixels in the camera image correspond to the same code in the target planes. Figure 3 (top right) shows the result of a reconstruction if we apply this initial matching directly.

3.2 Sub-pixel Matching

From the Gray code decoding we get an initial integer-valued estimate of the pixel matching. To get more accurate correspondences, this initialization has to be refined. Let $u(x, y)$ and $v(x, y)$ denote the coordinates of the target point corresponding to the camera pixel (x, y). Instead of directly smoothing u and v as in [13], we use an energy minimization approach to ensure that the smoothed correspondences will still link camera pixels with their corresponding origin on the target planes.

We minimize the following energy functional with respect to u and v:

$$\begin{aligned} E(u, v) = &\sum_k \int_\Omega (\mathcal{G}_k(u, v) - \mathcal{I}_k(x, y))^2 \, dx \, dy \\ &+ \lambda \int_\Omega |\nabla u|^2 + |\nabla v|^2 \, dx \, dy \end{aligned}$$

where Ω is the mirror image region, $\mathcal{G}_k$ are the Gray code images and $\mathcal{I}_k$ are the images captured by the camera.

The first energy term is the data term. It penalize correspondences for which the color $\mathcal{I}_k(x, y)$ captured by the camera and it's corresponding Gray code $\mathcal{G}_k(u, v)$ are not the same. We first scale the camera images intensities pixel-wise, so that 0 and 1 intensities correspond to pure black and pure white. This referential is computed by displaying entirely black and entirely white images on the planar targets. For non-integer values of u and v, $\mathcal{G}_k(u, v)$ is computed using bilinear interpolation.

The second term is a homogeneous regularizer. It penalizes large variations on the correspondence functions. The λ parameter sets the compromise between data evidence and smoothing.

The energy functional is minimized by a steepest descent. The descent direction is given by the Euler-Lagrange equations,

$$\frac{\partial u_i}{\partial t} = -\sum_k 2(\mathcal{G}_k - \mathcal{I}_k)\frac{\partial \mathcal{G}_k}{\partial u_i} + \lambda \, 2\Delta u_i$$

for $u_1 = u$ and $u_2 = v$.

Figure 3 (bottom right) shows the result of the reconstruction after having smoothed the orginal matches.

4 Target Pose Estimation

Our reconstruction algorithm requires knowledge of the relative pose between the camera and target plane, in its different positions.

The first and simplest method is to ensure that the target plane is partially visible in the camera, as seen in figure 3, and apply any pose estimation method [14]; we use the method proposed in [15].

To ensure a much higher flexibility, we wanted to be able to work with setups where the camera hasn't any direct view of the target plane; if this was possible then one would be able to take "better" images of the specular surface to be reconstructed. The second solution is to estimate the pose of the targets through the reflection by a known mirror. We therefore suppose having a means of estimating the pose of the planar mirror: this can either be done by placing markers on the mirror and performing a classical plane pose estimation, or in our case by using a hard-drive platter, whose known interior and exterior radii allow an ellipse based pose to be estimated. More details on the reflection by a known plane can be found in the next paragraph.

4.1 Pose Through Reflection by 3 Unknown Planes

We acquire images by holding a planar mirror in front of the camera in different positions, such that the target plane's reflection is seen by the camera. We now briefly describe how to solve the relative pose between camera and target plane, from three or more such images, or one image of three or more such mirrors.

In the following, we adopt a global reference frame such that the target plane is at $Z = 0$, and first carry out a pose estimation for each image, as if the image were a direct view of the target plane.

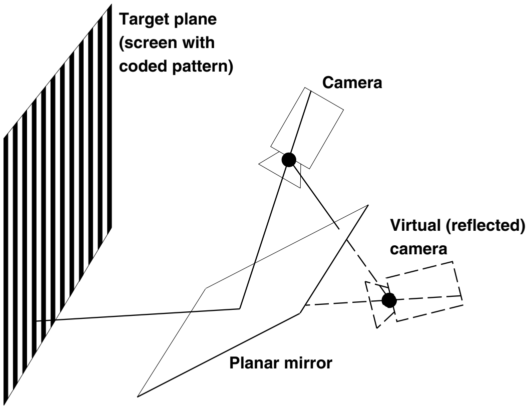

Target plane (screen with coded pattern)
Camera
Virtual (reflected) camera
Planar mirror

Fig. 2. Reflected pose. The estimated pose of a reflected plane is equivalent to its pose viewed from a virtual reflected camera.

This procedure gives us the pose of the virtual camera that would be produced by reflecting the real camera in the planar mirror, cf. figure 2. If we knew the pose of the planar mirror, we could of course immediately recover the camera's true pose, as follows. Let the recovered pose of the virtual camera for image i be given via the projection matrix:

$$\mathsf{P}_i^v \sim \mathsf{S}_i \left(\mathtt{I} | -\mathbf{t}_i\right)$$

where S_i is a reflection matrix (a rotation matrix multiplied by -1), and let the associated pose of the planar mirror be represented by homogeneous coordinates

$$\Pi_i \sim \begin{pmatrix} \mathbf{n}_i \\ d_i \end{pmatrix}$$

where we distinguish the plane's normal vector $\mathbf{n}_i$ (of unit norm), and its distance d_i from the origin. The true camera's pose can be recovered by multiplying P_i^v with the transformation modeling the reflection in the plane Π_i:

$$\begin{aligned} \mathsf{P}_i &\sim \mathsf{P}_i^v \begin{pmatrix} \mathtt{I} - 2\mathbf{n}_i\mathbf{n}_i^\mathsf{T} & -2d_i\mathbf{n}_i \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \\ &\sim \mathsf{S}_i \left(\mathtt{I} - 2\mathbf{n}_i\mathbf{n}_i^\mathsf{T} | -\mathbf{t}_i - 2d_i\mathbf{n}_i\right) \end{aligned} \tag{1}$$

We now have to address the question how to recover the true camera's pose, knowing that with the correct mirror positions Π_i, the camera poses P_i computed according to (1), have to be equal to one another: $\mathsf{P}_i \sim \mathsf{P}_j$. Due to $\det\left(\mathtt{I} - 2\mathbf{n}_i\mathbf{n}_i^\mathsf{T}\right) = \det \mathsf{S}_i = -1$, we can safely eliminate the scale ambiguity in the equation $\mathsf{P}_i \sim \mathsf{P}_j$, and obtain element-wise equalities:

$$\forall i,j : \mathsf{S}_i \left(\mathtt{I} - 2\mathbf{n}_i\mathbf{n}_i^\mathsf{T}\right) = \mathsf{S}_j \left(\mathtt{I} - 2\mathbf{n}_j\mathbf{n}_j^\mathsf{T}\right) \tag{2}$$

$$\forall i,j : \mathsf{S}_i \left(\mathbf{t}_i + 2d_i\mathbf{n}_i\right) = \mathsf{S}_j \left(\mathbf{t}_j + 2d_j\mathbf{n}_j\right) \tag{3}$$

Computing mirror plane normals $\mathbf{n}_i$. Let $\mathsf{X}_i = \mathtt{I} - 2\mathbf{n}_i\mathbf{n}_i^\mathsf{T}$, which is of course a symmetric matrix. From (2), we get:

$$\mathsf{X}_i = \underbrace{\mathsf{S}_i^\mathsf{T}\mathsf{S}_j}_{\mathsf{R}_{ij}} \mathsf{X}_j \tag{4}$$

Furthermore, X_j is a reflection, *i.e.* $\mathsf{X}_j\mathsf{X}_j = I$, therefore:

$$\mathsf{R}_{ij} = \mathsf{X}_i\mathsf{X}_j \tag{5}$$

Let $\mathbf{a}_{ij}$ be a vector orthogonal to $\mathbf{n}_i$ and $\mathbf{n}_j$. We therefore have:

$$\begin{aligned} \mathsf{R}_{ij}\mathbf{a}_{ij} &= \mathsf{X}_i\mathsf{X}_j\mathbf{a}_{ij} \\ &= \left(\mathtt{I} - 2\mathbf{n}_i\mathbf{n}_i^\mathsf{T}\right)\left(\mathtt{I} - 2\mathbf{n}_j\mathbf{n}_j^\mathsf{T}\right)\mathbf{a}_{ij} \\ &= \left(\mathtt{I} - 2\mathbf{n}_i\mathbf{n}_i^\mathsf{T}\right)\mathbf{a}_{ij} \\ &= \mathbf{a}_{ij} \end{aligned}$$

which implies that $\mathbf{a}_{ij}$ is the eigenvector to the eigenvalue 1 of R_{ij}, *i.e.* that $\mathbf{a}_{ij}$ is the rotation axis of R_{ij}.

We now have the means to compute all mirror normals $\mathbf{n}_i$, provided at least 3 mirrors are used.

1. Compute the pose eq. (1) of all virtual cameras, as described above.
2. For all pairs of mirrors (i, j), compute R_{ij}, as per eq. (4). Compute their eigenvectors to the eigenvalue +1, i.e. vectors $\mathbf{a}_{ij}$.
3. For every mirror i, stack all $\mathbf{a}_{ij}^{\mathsf{T}}$ (respectively $\mathbf{a}_{ki}^{\mathsf{T}}$) in a matrix A of size $(n-1) \times 3$ (where n is the number of mirrors), and compute $\mathbf{n}_i$ as the unit eigenvector to the smallest eigenvalue of $\mathsf{A}^{\mathsf{T}}\mathsf{A}$.

Computing the true camera's pose. The last step is to compute the least squares solution for the d_i of the linear equation system composed of one equation (3) per pair of mirrors. The system's design matrix is of size $3n(n-1) \times n$ and very sparse.

We now know all mirror planes Π_i, and can compute the camera pose from any one of them, according to eq. (1). In practice, we do this computation for every mirror, and then "average" the resulting rotation matrices and position vectors that represent camera pose. We then apply a bundle adjustment style procedure for simultaneously optimizing the pose of the camera and the planar mirrors. The cost function minimized here is the reprojection error of target points, projected in the camera after reflection in the mirrors.

5 Optimization

In practice, we also perform a global non-linear optimization of the poses T_1 and T_2 of the target planes, before the triangulation. The cost function to be minimized is the distance between matching lines in 3D space which we minimize using a Levenberg Marquardt algorithm.

$$cost(\mathsf{T}_1, \mathsf{T}_2) = \sum_{i \in \{matches\}} dist^2((\mathbf{O_c}, \mathbf{p_i}), (\mathsf{T}_1\mathbf{Q_{1i}}, \mathsf{T}_2\mathbf{Q_{2i}})).$$

6 Results

We tested our reconstruction method on real specular surfaces, using the different pose estimation methods presented in section 4. As seen on figure 3, no continuity or regularity is assumed.

Having no ground truth results, we evaluated the correctness of the method by fitting a plane to the part of the reconstruction we knew was planar, *i.e.* the hard drive platter (linear least squares fitting, without outlier removal). In the reconstruction shown on figure 3, over 98% of the computed points were less than 0.2 mm away from the surface, and 64% less than 0.1 mm. The approximate diameter of the reconstructed part of the platter was 80 mm, resulting in a maximum 0.3% relative error in the reconstruction.

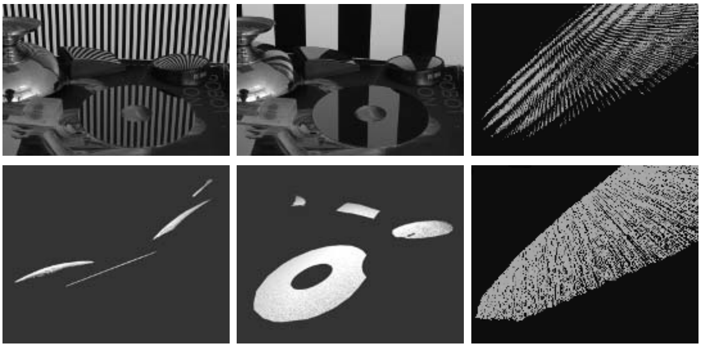


Fig. 3. Validation Setup and Results. The top row shows two of the images used for the reconstruction. Notice the 3 curved mirrors (an ice-cream cup and two small wide-angle rear-view mirrors, the planar hard drive platter, and a direct view of the target plane, in the upper part of the image. The second row shows the reconstruction viewed from two locations. The model contains over 525 000 independent points. Note the planarity of the reconstructed hard drive platter in the left image. Only a few points could be computed on the ice-cream cup, as its surface covered by the exploitable Gray codes was limited. The two small rear-view mirrors (one with circular, the other with rectangular based shape) were completely reconstructed (apart from a non specular dent in the circular one). The two images on the right show the effect of the sub-pixel matching and constrained smoothing: top image shows result using raw gray codes, while the bottom one shows results after the smoothing step.

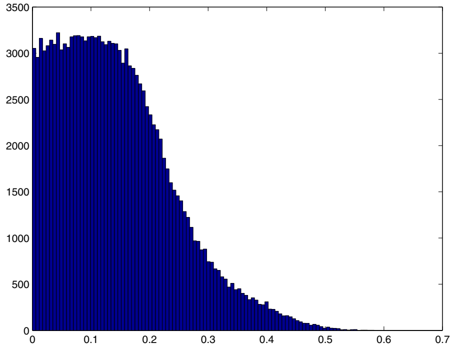


Fig. 4. Point-plane distance. Histogram of the distance in of each point to the linear least squares fitted plane (in millimeters) with the poses estimated with the three unknown mirror planes (section 4.1).

The accuracy of the reconstruction also depends on the quality of the pixel matching. Indeed, when experimenting with purely piecewise planar surfaces, where the sub-pixel matching was "easy" to compute, the distances to the fitted planes dropped down to 99.9% of the computed points less than 0.1 mm away from the surface, and 88% less than 0.05 mm. This is because the average quality

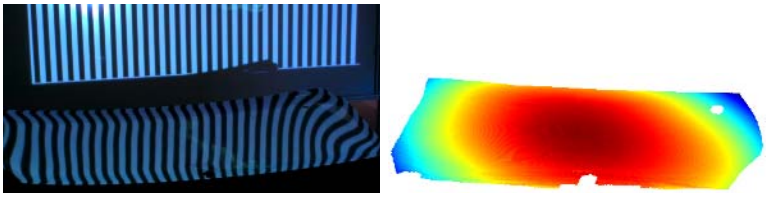


Fig. 5. Real World Reconstruction. Reconstruction of a car windshield. The method allowed us to easily obtain a 800 000 + point model using a classical video projector, on a large scale reflective surface. The hole in the middle is due to a non-specular patch on the surface.

of the matches is higher compared to when the scene also contains curved specular surfaces. Hence the pose of the target planes and finally the reconstruction are more precise.

We tested the reconstruction on another setup composed only of planes with the different pose estimation techniques presented in section 4. Although the initial estimation of the poses given by the different techniques are not exactly identical, the non-linear optimization converged to very similar poses in all cases. The histogram of the point-plane distance, with the poses estimated with the three unknown planes (section 4.1), without global optimization, can be seen in figure 4.

7 Conclusion

We have presented a novel method that reconstructs a specular surface from two views. Compared to other reconstruction methods, we attain a high level of accuracy, without having the need to suppose surface continuity or regularity. We believe it could easily be implemented in an industrial surface inspection application, at least to provide an accurate initialization for integration based reconstruction methods, probably the only purely vision based ones able to detect surface micro-structure. We also proposed a novel method for the pose estimation of a target plane even if it is never directly seen in the images, requiring the view of its reflection through unknown planar mirrors.

The drawback of the method is the need to obtain a dense matching over the *complete* surface we want reconstructed. This in practice is difficult to obtain with only two positions of the target plane, meaning multiple reconstructions have to be computed then stitched together.

References

1. Chen, F., Brown, G.M., Song, M.: Overview of three-dimensional shape measurement using optical methods. Optical Engineering **39** (2000)
2. Blake, A., Brelstaff, G.: Geometry from specularities. In: Second International Conference on Computer Vision (Tampa,, FL), Washington, DC,, Computer Society Press (1988) 394–403

3. Zisserman, A., Giblin, P., Blake, A.: The information available to a moving observer from specularities. Image and Vision Computing **7** (1989) 38–42
4. Oren, M., Nayar, S.K.: A theory of specular surface geometry. In: International Conference on Computer Vision. (1995) 740–747
5. Zheng, J.Y., Murata, A.: Acquiring 3D object models from specular motion using circular lights illumination. In: Procedings of the Sixth International Conference on Computer Vision (ICCV-98). (1998) 1101–1108
6. Halstead, M., Barsky, B., Klein, S., Mandell, R.: Reconstructing curved surfaces from specular reflection patterns using spline surface fitting of normals. In: SIGGRAPH 96 Conference Proceedings. (1996) 335–342
7. Tarini, M., Lensch, H., Goesele, M., Seidel, H.: 3D acquisition of mirroring objects. In: Research Report MPI-I-2003-4-001, Max-Planck-Institut fr Informatik (2003)
8. Solem, J.E., Aanæs, H., Heyden, A.: A variational analysis of shape from specularities using sparse data. In: 3DPVT, IEEE Computer Society (2004) 26–33
9. Savarese, S., Chen, M., Perona, P.: Recovering local shape of a mirror surface from reflection of a regular grid. In: European Conference on Computer Vision. (2004)
10. Bonfort, T., Sturm, P.: Voxel carving for specular surfaces. In: International Conference on Computer Vision. (2003) 591–596
11. Grossberg, M., Nayar, S.: A general imaging model and a method for finding its parameters. In: International Conference on Computer Vision. (2001) 108–115
12. Sturm, P., Ramalingam, S.: A generic concept for camera calibration. In: Proceedings of the European Conference on Computer Vision. Volume 2., Springer (2004) 1–13
13. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: CVPR (1), IEEE Computer Society (2003) 195–202
14. Haralick, R., Lee, C., Ottenberg, K., Nolle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. IJCV **13** (1994) 331–356
15. Sturm, P.: Algorithms for plane-based pose estimation. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA. (2000) 1010–1017

Dense 3D Reconstruction with an Uncalibrated Active Stereo System

Hiroshi Kawasaki[1], Yutaka Ohsawa[1], Ryo Furukawa[2], and Yasuaki Nakamura[2]

[1] Faculty of Engineering, Saitama University,
255, Shimo-okubo, Sakura-ku, Saitama, Japan
{kawasaki, ohsawa}@mm.ics.saitama-u.ac.jp
[2] Faculty of Information Sciences, Hiroshima City University,
3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima, Japan
{ryo-f, nakamura}@cs.hiroshima-cu.ac.jp

Abstract. In this paper, we describe a novel uncalibrated active stereo system using coded structured light. Structured-light-based active stereo systems generally consist of a camera and projector that require precise precalibration. Therefore, if we can eliminate the precalibration process from the system, the user can merely place the equipment arbitrarily and directly begin scanning the objects. This will greatly improve both the convenience and practicality of the system. In order to achieve this, we propose an original self-calibration method that can be considered as a camera-to-camera self-calibration method in which one of the cameras is replaced with a projector. We also propose a simultaneous 3D reconstruction method that utilizes multiple captured stereo pairs to increase the accuracy of the 3D estimation. Further, we suggest a simple solution to eliminate the ambiguity of scaling by attaching a laser pointer to the projector, which is important for the practical use of the 3D reconstruction.

1 Introduction

3D acquisition stereo systems can be categorized into two basic types: a passive stereo system and an active stereo system. The former can recover 3D shapes only from multiple images, therefore no special devices are necessary and the systems are usually easy to use. However, in order to recover 3D shapes from images by passive stereo, accurate correspondences between images are required, making this a difficult task.

On the other hand, an active stereo system utilizes a light or laser projector for scanning and can thus retrieve high-precision correspondences with ease; therefore, the accuracy of the 3D points is relatively high. Another benefit of this system is that dense 3D points can be captured easily by controlling the light-projecting devices. Among the different types of active stereo systems, the structured-light-based system is widely used because of its several advantages such as simplicity, efficiency of scanning , and inexpensiveness for production.

One of the serious drawbacks of active stereo systems is that they essentially require precalibration between the camera and projector whenever the system

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 882–891, 2006.

conditions are changed. Since the precalibration is usually complicated and laborious task, it significantly reduces the convenience of the system. If we can eliminate the precalibration process from an active stereo system, it will greatly improve both the convenience and practicality of the system.

On the bases of these facts, we propose an active stereo system that does not require precalibration. Our proposed method is based on a self-calibration stereo method that can be considered as a camera-to-camera self-calibrated method in which one of the cameras is replaced with a projector. The relative position and parameters of the projector model are estimated from the epipolar constraints between the camera and projector. The estimation is performed by applying Levenberg-Marquardt method to the correspondence points obtained by the coded structured light. We also propose several methods to eliminate the ambiguity of scaling, which is inevitable in uncalibrated 3D reconstruction methods.

The contributions of our work are as follows: the first is that our system does not assume any limitations on the shapes of objects in the scenes (e.g., the inclusion of planar surfaces or calibration objects) or on the camera models (e.g., assumption of an orthogonal camera model); the second is that we propose the simultaneous 3D reconstruction of multiple scans by which we can obtain constant scaling for multiple scenes and can greatly enhance accuracy; the third is that we propose a simple solution to eliminate the ambiguity of scaling by attaching a laser pointer to the projector, which is important for the practical use of 3D scanners.

2 Related Works

Many active 3D scanning systems have been proposed to date. Among them, a projector-camera based system is commonly used because of several advantages. However, the system requires a precise precalibration for installation and this is usually a laborious task.

In order to avoid the calibration problems mentioned above, many uncalibrated active stereo methods have been proposed [1, 2, 3]. Takatsuka [1] and Furukawa [3] have proposed active stereo 3D scanners with online calibration methods. Each of their systems consists of a video camera and a laser projector attached with LED markers, and executes projector calibration for every frame. Thus, system configuration is relatively free and a real-time system is achieved. However, when calibration is performed for each frame, the system tends to have insufficient accuracy and low efficiency for practical use.

Self-calibrated stereo techniques have been studied extensively with regard to passive stereo systems (i.e., camera-to-camera systems)[4], and several researchers have attempted to apply these techniques to active stereo systems by substituting one of the paired stereo cameras with a projector [5, 6]. Fofi et al. developed an active vision system with self-calibration of extrinsic parameters, however, they assume an affine camera model and require planes to be scanned in the target scenes. Chen et al. estimated relative pose between the camera and the projector using nine points on a plane, thus, their method also needs planes in the scenes.

3 Uncalibrated Active Stereo System

3.1 System Configuration

The 3D reconstruction system developed in this work consists of a video projector and a camera. A laser pointer is attached to the projector and is used for determining the scaling parameter, which cannot be estimated with uncalibrated stereo methods. If the ambiguity of the scaling parameter can be left unsolved, the laser pointer can be omitted. Fig. 1 shows the configuration of the system.

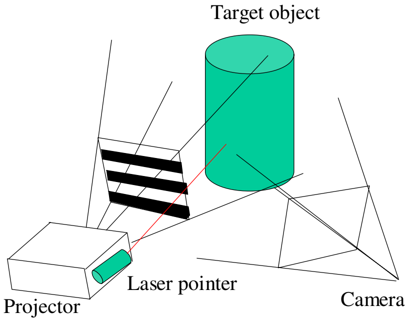


Fig. 1. Components of the 3D measurement system

The camera and projector are oriented toward the object to measure the shape. A set of dense correspondence points is obtained by the structured light method. The camera-projector parameters are self calibrated using the points. The 3D locations of the correspondence points are then reconstructed by using the stereo method.

Our proposed system has the following features, which are highly desirable in a practical 3D measurement system. Firstly, the projector and the camera can be located arbitrarily. Secondly, there are no limitations imposed on the geometry of the measured scene.

3.2 Obtaining a Set of Correspondence Points by Structured Light

To obtain correspondence points effectively by using video projector, coded structured light methods have been used and studied extensively[7, 8]. In the present method, directions from the projector are encoded into the light patterns, which are projected onto the target surface. The light patterns projected to each pixel are decoded from the obtained images, and the mapping from each pixel in the images to directions from the projector is obtained.

Since the light patterns encode 1D locations in the projected patterns, we applied the code twice, once for the x-coordinate of the projected pattern and once for the y-coordinate. Based on the compound light patterns, point-to-point correspondences between the directions from the projector and the pixels in the image are resolved.

3.3 Self-calibration and 3D Reconstruction

Our aim is to construct an active stereo system that does not require any calibration process even if the camera or projector is moved arbitrarily; therefore, the camera parameters should be self-calibrated. We assume that the intrinsic parameters of the camera and the projector are known, except the focal length of the projector. This is because the intrinsic parameters of the camera can be obtained easily by existing methods, while those of the projector are more difficult to obtain. Another reason is that, in our experience, focus of the projector must be adjusted more frequently than those of the cameras in actual scanning processes, due to the relatively shallow depth of field of the projectors. Therefore, we estimate the focal length of the projector and the extrinsic parameters of the relative position between the camera and projector by the self-calibration method.

For the self-calibration, a nonlinear optimization, Levenberg-Marquardt method, is applied. Recently, due to the improved computational capabilities of PCs, self-calibration and 3D reconstructions using only nonlinear optimizations have been studied [9], and this approach is employed in our study.

We call a coordinate system which is fixed with the projector (or the camera) the projector (camera) coordinate system. Coordinate values expressed in this system are the projector (camera) coordinate. The origin of the projector (camera) coordinate system is the optical center of the projector (camera). The forward direction of the projector (camera) is the minus direction of the z-axis of the projector (camera) coordinate system. The x and y-axis of the projector (camera) coordinate system are parallel with the vertical and horizontal directions of the image coordinate system of the screen.

Let the focal length of the projector be f_p, and the direction vector of the ith correspondence point expressed in the projector coordinates be $(u_{pi},\ v_{pi},\ -f_p)^t$.

Here, we express the rigid transformation from the projector coordinates to the camera coordinates as the rotation matrix $\mathbf{R}_p$ and the translation vector $\mathbf{t_p}$. The rotation is expressed by the parameters of Euler angles α_p, β_p and γ_p, and the rotation matrix is thus expressed as $\mathbf{R}_p(\alpha_p, \beta_p, \gamma_p)$. Since the norm of the translation vector $\|\mathbf{t_p}\|$ cannot be resolved by a self-calibration, $\mathbf{t_p}$ is assumed to be an unit vector and is expressed by two parameters of polar coordinates. Thus, $\mathbf{t_p}$ is expressed as $\mathbf{t_p}(\rho_p, \phi_p) := (\sin\phi_p \cos\rho_p, \sin\phi_p \sin\rho_p, -\cos\phi_p)^t$.

The direction of the correspondence points observed by the camera is converted to the screen coordinates of a normalized camera, with corrected effects of the lens distortions. Let the converted coordinates be $(u_{ci},\ v_{ci},\ -1)^t$.

If the epipolar constraints are met, the lines of sights from the camera and the projector intersect in the 3D space. The line from the projector in the camera coordinates is

$$r\{\mathbf{R}_p(\alpha_p, \beta_p, \gamma_p)\}(u_{pi}/f_p, v_{pi}/f_p, -1)^t + \mathbf{t_p}(\rho_\mathbf{p}, \phi_\mathbf{p}) \tag{1}$$

where r is an arbitrary value. The line from the camera is expressed as $s(u_{ci}, v_{ci}, -1)^t$ where s is an arbitrary value.

To achieve the epipolar constraints, the distance between the two lines should be minimized. Let the direction vectors of the lines be expressed as

$$\begin{aligned} \mathbf{p}_{ci} &:= N\ (u_{ci},\ v_{ci},\ -1)^t, \\ \mathbf{q}_{ci}(\theta, f_p) &:= N\ \{\mathbf{R}_p(\alpha_p, \beta_p, \gamma_p)\}(u_{pi}/f_p, v_{pi}/f_p, -1)^t, \end{aligned} \tag{2}$$

where N is an operator which normalizes a vector (i.e. $N\ \mathbf{x} := \mathbf{x}/\|\mathbf{x}\|$), and $\theta := (\alpha_p, \beta_p, \gamma_p)$ represents the parameters of rotation of the projector. Then, the signed distance between the lines is

$$E_i(\theta, \tau, f_p) := \mathbf{t_p}(\tau) \cdot N\ (\mathbf{p}_{ci} \times \mathbf{q}_{ci}(\theta, f_p)), \tag{3}$$

where "·" indicates dot product, and $\tau := (\rho_p, \phi_p)$ represents the parameters of the translation.

$E_i(\theta, \tau, f_p)$ includes systematic errors whose variances change with the parameters (θ, τ, f_p) and the data index i. To compose an error evaluation function unbiased about the parameters (θ, τ, f_p), $E_i(\theta, \tau, f_p)$ should be normalized by the expected error level. Assuming the epipolar constraints are met, the distance from the intersection of the lines to the camera and the projector are

$$\begin{aligned} D_{ci}(\theta, \tau, f_p) &:= \|\mathbf{t_p}(\tau) \times \mathbf{q}_{ci}(\theta, f_p)\| / \|\mathbf{p}_{ci} \times \mathbf{q}_{ci}(\theta, f_p)\|, \\ D_{pi}(\theta, \tau, f_p) &:= \|\mathbf{t_p}(\tau) \times \mathbf{p}_{ci}\| / \|\mathbf{p}_{ci} \times \mathbf{q}_{ci}(\theta, f_p)\|. \end{aligned} \tag{4}$$

Let the angle between the line of sight and the optical axis of the camera be ψ_c. Also, let the angle between the epipolar plane and the optical axis of the camera be ω_c. Let ψ_c and ω_c be defined similarly for the projector. These angles can be easily calculated.

Using the distances and angles, the signed distance normalized by the error level is expressed by $\tilde{E}_i(\theta, \tau, f_p)$ in the forms

$$\begin{aligned} w_i(\theta, \tau, f_p) &:= \{\epsilon_c \cos\psi_c \cos\omega_c D_{ci}(\theta, \tau, f_p) + \epsilon_p \cos\psi_p \cos\omega_p D_{pi}(\theta, \tau, f_p)/f_p\}^{-1} \\ \tilde{E}_i(\theta, \tau, f_p) &:= w_i(\theta, \tau, f_p)\ E_i(\theta, \tau, f_p) \end{aligned} \tag{5}$$

where ϵ_c and ϵ_p are the errors intrinsic to the camera and the projector expressed as lengths in the normalized screens. In our experiments, we used pixel sizes for ϵ_c and ϵ_p.

Then, the function $f(\theta, \tau, f_p)$ to be minimized with the non-linear optimization is expressed as the following form:

$$f(\theta, \tau, f_p) := \sum_{i=1}^{K} \{\tilde{E}_i(\theta, \tau, f_p)\}^2, \tag{6}$$

where K is the number of correspondences. The function is minimized using Levenberg-Marquardt method.

Once we obtain the parameters t_p and R_p , we can directly recover the 3D shapes by the stereo method.

3.4 Simultaneous Reconstruction of Multiple Scenes

Since the ambiguity of scaling inevitably exists in uncalibrated stereo methods, several problems occur in practical use. For example, when we scan an object from various view directions to capture its entire shape, different scaling parameters for each scan make it difficult to achieve correct registration and integration.

The simultaneous 3D reconstruction is performed as follows. First, multiple scenes are captured by keeping both the camera and projector fixed. The intrinsic and extrinsic parameters of the camera and projector are identical for all the scenes. Therefore, by merely joining the sets of correspondence points for all the scenes and applying the self-calibration algorithm described in section 3.3, consistent camera and projector parameters are obtained. The 3D reconstruction can then be performed for each scene using the estimated camera/projector parameters.

An advantage of this method is that the scalings of all the reconstructed scenes are the same because of the use of consistent camera and projector parameters for multiple scenes. This simplifies the problem of registration and integration for capturing the entire shape of an object.

3.5 Estimation of Scaling Parameter

A reconstructed 3D shape produced by our method is scaled by an unknown multiplier from the real shape. The simultaneous reconstruction described in the previous subsection is useful for obtaining multiple reconstructions with constant scaling, but we cannot estimate "real" factor of scaling for the scene with the method. For some applications, estimation of the real scaling factor is needed.

To achieve this, the following methods can be applied to determine the multiplier: which are

(1) measuring the length of the two points on the real shape,
(2) measuring an object with a known shape (a calibration object) and the target object successively without moving the camera nor the projector,
(3) or measuring a calibration object and the target object simultaneously.

However, all of these techniques normally require some human intervention such as measuring or specifying the calibration object, making it difficult to develop a completely automatic measuring process.

To determine the scaling parameters more easily, we attach a laser pointer to the projector and project a mark onto the measured surface, which is then observed by the camera. The projected laser light forms a fixed line in the 3D space expressed by the projector coordinates. From the 3D point lit by the pointer in the image, the line of sight to the point is determined. The scaling parameter is calculated by applying triangulation to the line of the laser and the line of sight.

The line formed by the laser pointer should be calibrated in the projector coordinate system. To do this, multiple points on the laser are obtained by measuring an object with a known shape lit by the laser line. The points are fitted to a line to obtain the parameters of the laser line. Calibration is needed only once when the laser pointer is attached to the projector.

4 Experiments

4.1 Evaluation of Accuracy

To evaluate our proposed method, we scanned a scene of a cube (20cm × 20cm × 20cm) (Fig.2(a)and(b)), calculating the extrinsic parameters and the focal length of the projector. To evaluate the effectiveness of our simultaneous 3D reconstruction method, we performed simultaneous 3D reconstructions and single-scene 3D reconstructions and compared the results. For a comparison, we also performed an explicit calibration of the extrinsic parameters and the focal length of the projector using the known 3D positions of the markers on the cube as the ground truth.

For the test data, we scanned a cube-shaped object 5 times changing the position of the object. During the scanning, we kept the camera and projector fixed. Then,3D reconstruction was performed for each scanned data, which was referred as a single-scene reconstruction. Simultaneous reconstruction was performed as follows. First, self-calibration was done using all the scanned data as a single input. Then, using the estimated parameters, each 3D shape was reconstructed by using the stereo method. Each of the self-calibrations was performed under 2 conditions: one was a self-calibration with fixed focal length of the projector and the other included the estimation of the focal length. For the fixed focal length condition, the focal length calculated by the explicit calibration was used. The initial values of the position and direction of the projector were α_p=0°, β_p=20°, γ_p=0°, $\mathbf{t_p}$=(1,0,0), $f_p = 0.05$. The estimated parameters are shown in Tab. 1.

We also evaluated the accuracy of the obtained 3D point set shown in Fig.2(c), (d). We applied a plane fitting algorithm for the faces in the scene, obtaining 3 planes (A,B,C) shown in Fig.2(a). The fitting of planes to the point sets were

Table 1. Parameters estimated by calibration and from data

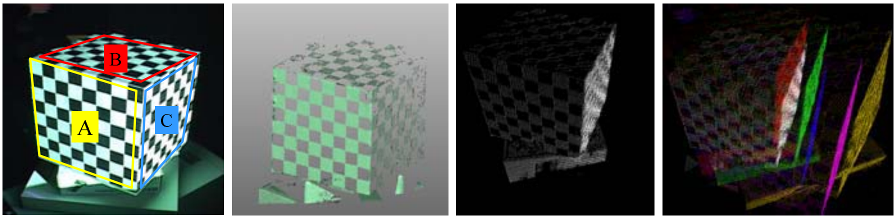
	By calibration	From data
f_p	0.0338[m]	0.0329[m]
$(\alpha_p, \beta_p \gamma_p)$	(-9.3°, -31.6°, -13.0°)	(-8.2°, -30.9°, -12.7°)
$\mathbf{t_p}/\|\mathbf{t_p}\|$	(-0.610,0.446,-0.655)	(-0.581,0.441,-0.684)

Table 2. Results of angles between estimated planes

	Single Input		Simultaneous	
Focus of projector	Fixed	Selfcalib	Fixed	Selfcalib
between A-C	89.95°	90.23°	90.17°	90.06°
between B-C	89.87°	90.99°	89.88°	90.56°
between A-B	90.39°	92.09°	90.07°	91.33°

Table 3. Summary of 3D reconstruction

	Single Input		Simultaneous	
Focus of projector	Fixed	Selfcalib	Fixed	Selfcalib
Ave. errors	0.02°	2.65°	-0.01°	-0.52°
RMSE (degree)	0.10 °	3.78°	0.07°	1.79°
RMSE of plane(mm)	0.72	0.77	0.62	0.65



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

Fig. 2. Scanning of a cube with known size: (a) 3 faces used for accuracy estimations, (b) reconstructed 3D points, (d) 3D point set acquired by single-scene reconstruction, and (d) by simultaneous reconstruction

performed by principal component analysis. By using the estimated plane parameters, we calculated angles between the estimated planes. One of the measured angles of the 5 data sets are shown in Tab. 2. The averaged signed errors from the actual angles (90 °)and the roots of mean squared errors (RMS errors) of them are shown in Tab. 3.

Also, the residual RMS errors of the plane fitting algorithms were calculated, which are shown in Tab. 3.

From the results, we can see that the shapes of the cubes are correctly reconstructed for all of the conditions. From the tendency of the results, we can see that the results of the simultaneous reconstructions were better than the single reconstructions for both of the accuracy of angles and residuals of plane-fittings. Thus, these experimental results show the effectiveness of the simultaneous reconstructions.

The results of the calibrated focal lengths were slightly worse than those of fixed focal lengths, but they had sufficient accuracy for practical use. One possible reason for the estimation errors might be inappropriate error model used in our system.

4.2 Scaling Parameter Evaluation

We evaluated our method for estimating the scaling parameter. First, the line formed by the laser light was calibrated by measuring 2 points lit by the laser. Then, we moved the camera and projector, scanned a cube with the initial values $(\alpha_p, \beta_p, \gamma_p)$=(0°, 20°, 0°), $\mathbf{t_p}$=(1,0,0), $f_p = 0.05$, and calculated the scale of the measured point set from the data. To evaluate the accuracy of the estimated scaling factor, we measured the length of 3 edges of the cube, a, b, and c. The results were 199.9mm, 199.2mm and 199.8mm, respectively. All estimated length of edges were close to the actual length 200.0mm, thus confirming that our estimation method is effective and practical.

4.3 Entire Shape Acquisition by Simultaneous Method

To demonstrate the effectiveness of our simultaneous 3D reconstruction method, we performed entire shape acquisitions of two objects, a china figurine and a helmet.

In order to reconstruct entire shapes, we scanned each of the objects 8 times, rotating it by 45°. Then, all 3D shapes were recovered simultaneously by our simultaneous 3D reconstruction method. Finally, we appled an alignment algorithm for shape registration.

Results are shown in Fig.3. We can observe that multiple scanned shapes are integrated into a single shape without any gaps, although the system is uncalibrated. This is because all the scaling parameters for 8 scan data sets are the same, which is achieved by the simultaneous reconstruction method.

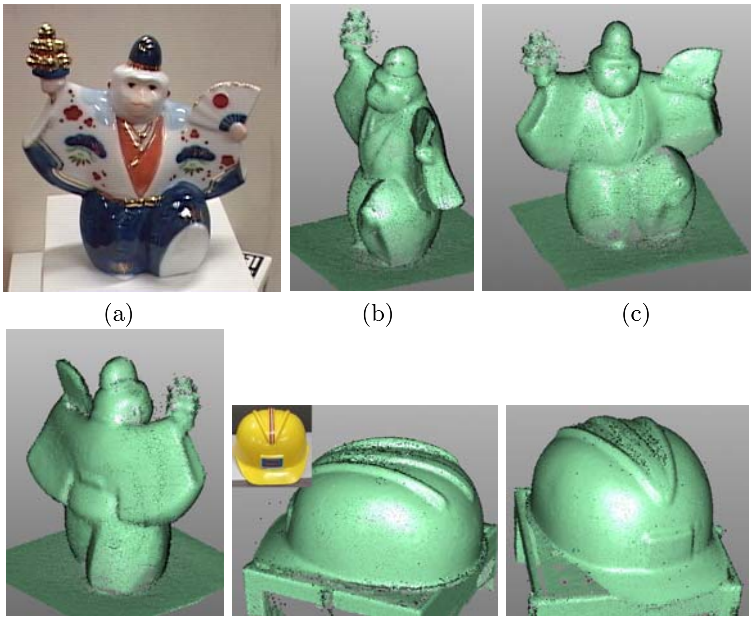


Fig. 3. Examples of the scanned objects: (a)(b)(c)(d) a china figurine, (e)(f) a helmet

5 Conclusion

In this paper, we propose a novel uncalibrated active stereo system that enables dense 3D scanning with a single scanning process and without any precise calibrations or special devices. Our proposed method is based on an uncalibrated stereo technique for a passive stereo system in which one of the cameras is replaced with a projector. We also propose a simultaneous 3D reconstruction method to increase accuracy and a simple method to eliminate the ambiguity of scaling by attaching a laser pointer to the projector.

By using our proposed method, the camera and projector can be arbitrarily installed and it is possible to start 3D scanning immediately and without any precalibrations or complicated preparations. To verify the reliability and effectiveness of our proposed method, we conducted several experiments with the proposed system and actual objects. The results of our experiments confirm the effectiveness of our proposed system.

References

1. Takatsuka, M., West, G.A., Venkatesh, S., Caelli, T.M.: Low-cost interactive active monocular range finder. In: CVPR. Volume 1. (1999) 444–449
2. Davis, J., Chen, X.: A laser range scanner designed for minimum calibration complexity. In: Third Int. Conf. on 3DIM. (2001) 91–98
3. Furukawa, R., Kawasaki, H.: Interactive shape acquisition using marker attached laser projector. In: Int. Conf. on 3DIM2003. (2003) 491–498
4. Faugeras., O.: Three-Dimensional Computer Vision - A Geometric Viewpoint. Artificial intelligence. M.I.T. Press Cambridge, MA (1993)
5. Fofi, D., Salvi, J., Mouaddib, E.M.: Uncalibrated vision based on structured light. In: ICRA. (2001) 3548–3553
6. Chen, S.Y., Li, Y.F.: Self-recalibration of a colour-encoded light system for automated three-dimensional measurements. Measurement Science and Technology **14** (2002) 33–40
7. Caspi, D., Kiryati, N., Shamir, J.: Range imaging with adaptive color structured light. IEEE Trans. on Patt. Anal. Machine Intell. **20** (1998) 470–480
8. Inokuchi, S., Sato, K., Matsuda, F.: Range imaging system for 3-D object recognition. In: ICPR. (1984) 806–808
9. Amano, A., Migita, T., Asada, N.: Stable recovery of shape and motion from partially tracked feature points with fast nonlinear optimization. In: 15th Vision Interface. (2002) 244–251

Surface-Independent Direct-Projected Augmented Reality

Hanhoon Park, Moon-Hyun Lee, Sang-Jun Kim, and Jong-Il Park

Division of Electrical and Computer Engineering,
Hanyang University, Seoul, South Korea
{hanuni, vivendi, markjun}@mr.hanyang.ac.kr,
jipark@hanyang.ac.kr
http://mr.hanyang.ac.kr

Abstract. Some issues on direct-projected augmented reality (DirectAR) are addressed: the projection may be geometrically distorted due to the non-planar surface (geometric distortion); the projection cannot be seen to user as intended because the position of the projector is not the same as that of the user's viewpoint (viewpoint-ignorant projection); the projection may be modulated by surface color (radiometric distortion); the projected area may not have uniform brightness when the projection is obliquely headed for the surface (uneven projection). We propose an integrated framework for handling all the problems. Experimental results demonstrate that the problems unavoidable in surface-independent DirectAR can be successfully resolved.

1 Introduction

Direct-projected augmented reality (hereafter, it is called DirectAR) approaches have been proposed and applied to many applications such as enhancing the face of an actor or changing the color and texture of real objects [12]. Projection made it possible to use 3-D real and large objects as displays [7] and freed from discomforts incidental to wearing a device such as HMD. For instance, surgeons schedule the operation and check up the state of patient while seeing magnetic resonance imaging (MRI) or computed tomography (CT) images. Medical image visualization has been proved to be useful since the methods for 3-D reconstruction and visualization of the MRI or CT images were emerged [1]. However, it is still stressful for surgeons to keep peering at the CRT display or wearing HMD to see the information during operation. By directly projecting the 3-D reconstructed MRI or CT images onto the patient's body, surgeons can be visually assisted in such a way that they can operate and monitor the patient's state simultaneously.

However, DirectAR usually suffers from the following problems: the projection may be geometrically distorted due to the non-planar surface (*geometric distortion*); the projection cannot be seen to user as intended because the position of the projector is not same as that of the user's viewpoint (*viewpoint-ignorant projection*); the projection may be modulated by the surface color (*radiometric distortion*); the projected area may not have uniform brightness when the projection is obliquely headed for the surface (*uneven projection*). Some papers have addressed these problems partially [8, 11]. In this paper, we aim at providing an integrated framework for handling all the problems.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 892 – 901, 2006.

Recently, multi-projector-based methods have been proposed to resolve the problems such as multi-focal projection [14] and specularity-free projection [15] other than the aforementioned problems. In this paper, we focus only on the problems regarding using a single projector.

2 Method

In this paper, we attempt to provide practical and easy-to-use algorithms coping with the problems of DirectAR. There is still room for improving the accuracy and further exploration.

2.1 Geometric Registration

Geometric registration indicates that projection is exactly overlaid on the surface without distortion. To do so, the projectors should be calibrated and the surface geometry should be known first. We use a modified version of the well-known Zhang's calibration method [6] for calibrating projectors and a linear triangulation method [5] for recovering the geometry of projection surface. The surface is triangularly represented using the recovered points on the surface because the surface is assumed to be piece-wise planar. Finally, the projection images are patch-wise prewarped using homography to be undistorted.

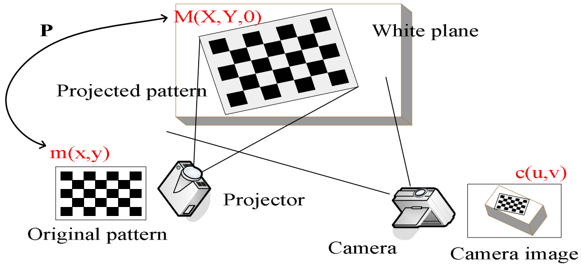


Fig. 1. Projector calibration

2.1.1 Projector Calibration

For calibrating a projector, a modified version of the well-known Zhang's calibration method [6] is used. Coplanar 3-D points and their corresponding 2-D points are required in the Zhang's calibration method. In our method, the points **m**(x,y) on a source pattern of a projector correspond to 2-D points and the projected points **M**(X,Y,0) correspond to 3-D points (see Fig. 1). The coordinates of the projected points are computed from the image captured by a camera as follows[1].

$$\begin{pmatrix} X & Y & 1 \end{pmatrix}^T = H_{c-o}\tilde{c} \qquad (1)$$

where H_{c-o} is camera-to-surface homography and $\tilde{c}$ is the homogeneous coordinates of the camera image. The relationship between 3-D points (**M**) and 2-D points (**m**) is represented in homogeneous coordinates as

[1] In this paper, tilde indicates homogenous coordinates.

$$\tilde{m} = P\tilde{M} = H_{o-p}\begin{pmatrix}X & Y & 1\end{pmatrix}^T \tag{2}$$

where H_{o-p} is surface-to-projector homography. Given the 3-D and 2-D points, optimization algorithms of Zhang's method are used as it is [6].

2.1.2 3-D Surface Modeling

Assuming that the camera and projector are calibrated, a linear triangulation method [5] is used for recovering the geometry of projection surface. The relationship between the coordinates of projection surface, projector coordinates, and camera coordinates is represented as

$$\tilde{m} = P\tilde{M}\ ,\ \tilde{c} = P_c\tilde{M}\ . \tag{3}$$

The homogeneous scale factor is eliminated by a cross product as

$$\tilde{m}\times(P\tilde{M}) = 0,\ \tilde{c}\times(P_c\tilde{M}) = 0 \tag{4}$$

where P_c is camera projection matrix. An equation of the form $AM = 0$ can then be composed, with

$$A = \begin{bmatrix}xp^{3T} - p^{1T} & yp^{3T} - p^{2T} & up_c^{\ 3T} - p_c^{\ 1T} & vp_c^{\ 3T} - p_c^{\ 2T}\end{bmatrix}^T \tag{5}$$

where p^{iT} are the rows of P. This is a redundant set of equations, since the solution is determined only up to scale. **A** has a 1-dimensional null-space which provides a solution for **M** and can be computed using Singular Value Decomposition (SVD) [5].

The surface is triangularly represented using the recovered points on the surface because the surface is assumed to be piece-wise planar. In practice, we project a grid pattern onto the surface and compute the 3D coordinates of only the corner points. A dense grid pattern can be used to model the complicated surface.

2.2 Radiometric Compensation

The color of projection is dependent on that of the projection surface. In other words, if the color of the surface is not pure white, the projection is modulated by the color of the surface. The radiometric compensation is a technique that makes the color of projection look unchanged by adjusting the color of the projection in advance when the projection surface has colorful texture. Letting **I** be the projector input image, the projected image $\mathbf{I_P}$ is acquired by projector response function *f* as

$$I_P = f(I)\,. \tag{6}$$

Thus, the projector input image $\hat{I}$ should be compensated by f^{-1} in advance such that

$$\hat{I} = f^{-1}(I_P)\,. \tag{7}$$

In this paper, the color change of projection is observed by a camera. The compensation result may be incomplete due to the unknown response function of the camera. However, the response function of camera can be easily estimated from multiple images captured with different exposure time [10].

The procedure of radiometric compensation consists of six steps in this paper.

Step 1: The geometric mapping between a projector and a camera is computed. It is explained in Section 2.1.

Step 2: The radiometric model of the pipeline from input projector color to the measured camera color is defined as

$$C = VI_P + F \quad \begin{array}{l} \textit{where } C\textit{: camera image point} \\ V\textit{: color mixing matrix} \\ I_P\textit{: projected image point} \\ F\textit{: ambient light} \end{array} \tag{8}$$

The **V** matrix captures all the coupling between the projector and camera channels and their interactions with the spectral reflectance [4].

Step 3: **F** using black projector image is computed.

Step 4: $\hat{V} = VD^{-1}$ is computed using one reference image and three images which have different values in R, G, B channel, respectively. **D** is the diagonal matrix with diagonal entries of **V** [4]. The recovery of $\hat{V}$ thus allows us to decouple the color channels. It means that C_k(k=r,g,b) is determined by only the input brightness of the channel k in Eq. (8). The entries of **D** can be computed from the linear equations which are given by multiplying Eq. (8) by $\hat{V}^{-1}$.

Step 5: The inverse response function f^{-1} of a projector is computed by comparing the 26 projector images with different gray-level i.e. I = 0, 10, 20, …, 250 with the 26 camera images which are acquired by capturing the projector images. f^{-1} is defined as 4th-order polynomial function in this paper because the plot of input projector brightness vs. camera image brightness has the shape as shown in Fig. 2. This computation is performed pixel-wise and channel-wise because each pixel has different response function and each channel also has different response function.

Step 6: The compensated projector input image $\hat{I}$ is computed by Eq. (7).

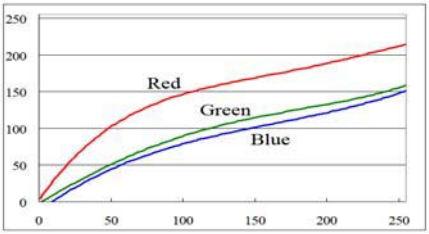


Fig. 2. Plot of projector input brightness (x-axis) vs. camera image brightness (y-axis). The curves are 4th- order approximation.

2.3 Viewer-Dependent Projection

For coping with the user's viewpoint which is tracked using an optical tracker [9] (see Fig. 3), it is assumed that the geometric transformation between user, projectors, and display surface is recovered and 3-D coordinates of the display surface is known. This problem has already been addressed in the previous section.

Figure 3 shows an AR-assistant surgery system [2] as an example of DirectAR system. The system visualizes the 3D position of tumor and additional information on the surface of human body. In the viewpoint of the projector, the direction of the

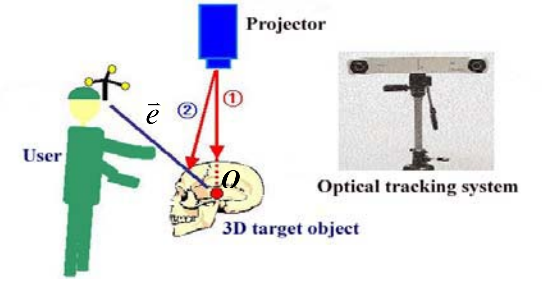


Fig. 3. DirectAR-based surgery system considering user's viewpoint. The graphical contents should be projected to not ① but ②.

projection should be ①. However, it is clear that the direction of the projection should be changed into ② when considering user's viewpoint.

To do so, the intersection point between $\vec{e}$ and the 3-D target object should be estimated. In this paper, a ray/triangle intersection algorithm is used [3]. The algorithm is divided into 2 steps. First, it is estimated if a ray is intersected with a certain triangle. Next, the coordinates of the intersection point is computed.

Step 1: Let V_i for $i \in$ 0,1,2 be the coordinates of the three vertices of the triangle. The normal vector of the triangle is represented by

$$\vec{n} = (V_1 - V_0) \times (V_2 - V_0)\,. \tag{9}$$

Any point V in the triangle's plane satisfies $V \cdot \vec{n}$ + *constant* = 0. The constant d is computed by

$$d = -V_0 \cdot \vec{n}\,. \tag{10}$$

If a ray parameterized by $O + \vec{e}\ t$ is intersected with a triangle, t parameter is computed by

$$t = (d - \vec{n} \cdot O)/(\vec{n} \cdot \vec{e})\,. \tag{11}$$

When $0 \le t \le 1$, the ray intersects with the triangle.

Step 2: A point V in the triangle plane is defined by

$$\overrightarrow{V_0V} = \alpha\overrightarrow{V_0V_1} + \beta\overrightarrow{V_0V_2} \quad \text{where } \alpha \ge 0, \beta \ge 0, \alpha + \beta \le 1. \tag{12}$$

In the image plane, this can be written as

$$\overrightarrow{v_0v}(u,v) = \alpha\overrightarrow{v_0v_1}(u_1,v_1) + \beta\overrightarrow{v_0v_2}(u_2,v_2) \tag{13}$$

where v_i is an image point of V_i. Therefore, α and β are computed by

$$\alpha = \det\begin{pmatrix} u_0 & u_2 \\ v_0 & v_2 \end{pmatrix} / \det\begin{pmatrix} u_1 & u_2 \\ v_1 & v_2 \end{pmatrix}, \quad \beta = \det\begin{pmatrix} u_1 & u_0 \\ v_1 & v_0 \end{pmatrix} / \det\begin{pmatrix} u_1 & u_2 \\ v_1 & v_2 \end{pmatrix}. \tag{14}$$

After computing the intersection point, the graphical contents are properly pre-warped and projected to the point without geometric distortion.

2.4 Intensity-Compensated Projection

When the projection is obliquely headed for the nonplanar surface, the area of the projection has uneven brightness as shown in Fig. 4. In this paper, we present a geometry-based method for making the projected area have even brightness. The target brightness is obtained by the average brightness of the projection area.

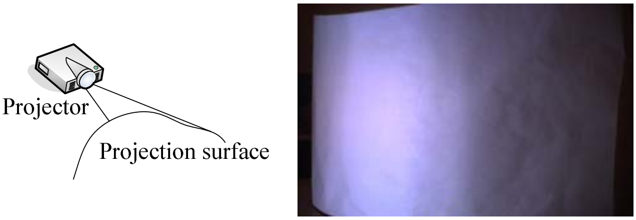


Fig. 4. Uneven projection. The projector lights the nonplanar surface in the left side. The projection is not uniform although an unicolored projection is applied to the surface.

The intensity of projection is dependent on the angle θ between the projection vector and the normal vector (see Fig. 5). The intensity is compensated as follows.

$$\hat{I}_i = wI_i, \quad i = r, g, b. \tag{15}$$

where

$$w = 1 + k_c \sin(\theta - \theta_p) \quad \textit{where } k_c\textit{: constant regarding the surface material} \tag{16}$$

θ_p: reference angle

Here, θ is computed as

$$\theta = \cos^{-1}(-\vec{n} \cdot \vec{p}), \qquad 0^\circ \le \theta \le 90^\circ. \tag{17}$$

This sinusoidal model for compensation was heuristically employed because it showed best performance in the various experiments.

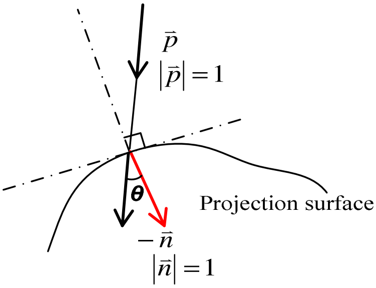


Fig. 5. Projection on nonplanar surface. θ represents the angle between two vectors. The intensity of projection per unit area is determined by θ.

The intensity of projection to the darker area is increased while the intensity of projection to the brighter area is decreased. Thus, the overall brightness of projection is similar to the brightness in the reference angle θ_p.

3 Experimental Results and Discussion

A projector (SONY VPL-CX6) and a camera (PointGrey Dragonfly) were used in our experiments. The images were at a resolution of 1024 by 768 pixels. An optical tracker (NDI Polaris) was used in the viewer-dependent projection.

Figure 6 shows the result of correcting the geometric distortion using the geometric registration method. After the projector was calibrated and the geometry of the cylindrical surface was recovered, the projection was prewarped to be undistorted.

Figure 7 shows the result of viewer-dependent projection. In the viewpoint of the projector, the projection is correct in Fig. 7-(b). However, the projection is wrong when considering the user's viewpoint. It was corrected and projected as shown in Fig. 7-(d) using the method explained in Section 2.3.

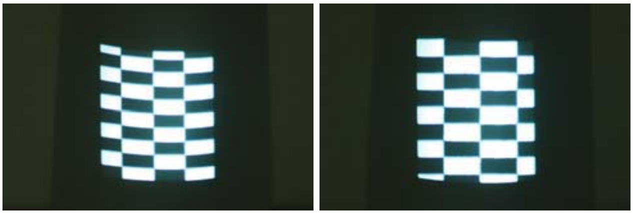
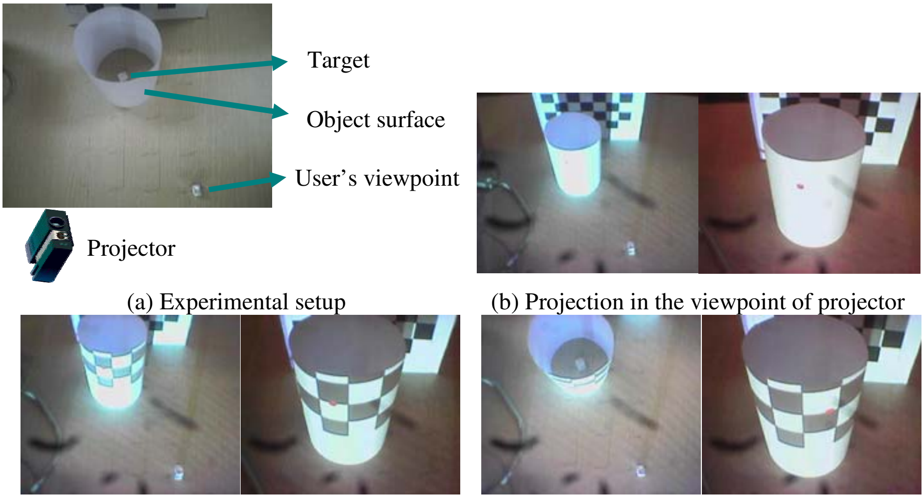


Fig. 6. Geometric distortion correction. Left image: before correction. Right image: after correction.



(a) Experimental setup

(b) Projection in the viewpoint of projector

(c) Projecting a grid pattern for surface modeling

(d) Projection in the viewpoint of user

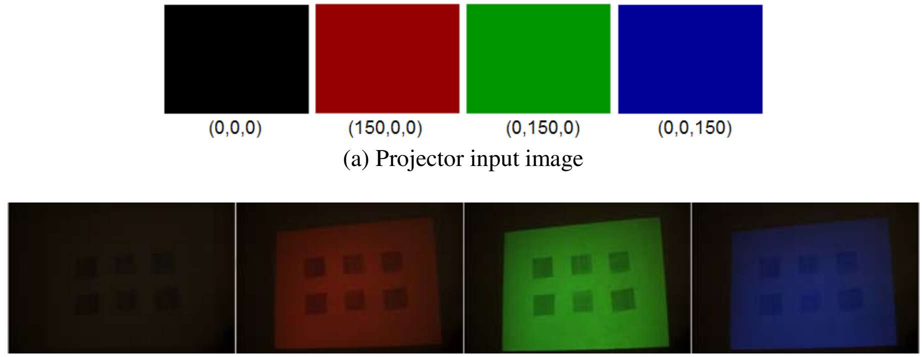
Fig. 7. Viewer-dependent projection

For radiometric compensation, the geometry of projection surface was recovered using the geometric registration method first. To compute $\hat{V}$, four images were used in Fig. 8. The black image is required to estimate **F**. Let I_0, I_r, I_g, I_b be the pixel of each camera image, respectively. Then, the entries v_{ij} of $\hat{V}$ have the values as

$$v_{11} = v_{22} = v_{33} = 1.0,$$
$$v_{12} = (I_g(\mathrm{R})\text{-}I_0(\mathrm{R}))/(I_g(\mathrm{G})\text{-}I_0(\mathrm{G})),\ v_{13} = (I_b(\mathrm{R})\text{-}I_0(\mathrm{R}))/(I_b(\mathrm{B})\text{-}I_0(\mathrm{B})),$$
$$v_{21} = (I_r(\mathrm{G})\text{-}I_0(\mathrm{G}))/(I_r(\mathrm{R})\text{-}I_0(\mathrm{R})),\ v_{23} = (I_b(\mathrm{G})\text{-}I_0(\mathrm{G}))/(I_b(\mathrm{B})\text{-}I_0(\mathrm{B})),$$
$$v_{31} = (I_r(\mathrm{B})\text{-}I_0(\mathrm{B}))/(I_r(\mathrm{R})\text{-}I_0(\mathrm{R})),\ v_{32} = (I_g(\mathrm{B})\text{-}I_0(\mathrm{B}))/(I_g(\mathrm{G})\text{-}I_0(\mathrm{G})).$$

The channel-wise and pixel-wise response function of the projector was estimated from 26 projector input images with different values and their camera images as mentioned in Section 2.2. Figure 2 shows an example of the estimated projector response function of a point. The coefficients c_i (i=0,1,2,3,4) of the blue-channel response function were as follows.

$$c_0 = \text{-}27.6826,\ c_1 = 2.7801,\ c_2 = \text{-}0.0165,\ c_3 = 3.3769\text{e-}005,\ c_4 = 1.2253\text{e-}008$$



(a) Projector input image

(b) Camera image (black) (c) Camera image (red) (d) Camera image (green) (e) Camera image (blue)

Fig. 8. Estimating the color mixing matrix $\hat{V}$

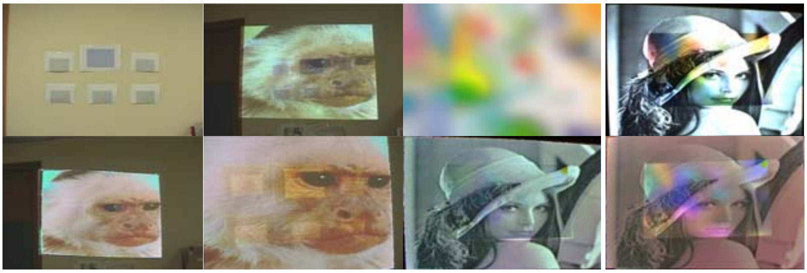


Fig. 9. Radiometric compensation. The right-top image shows that the projection color was distorted due to the texture of projection surface. There is no color distortion in the left-bottom image by compensating the projector input image. The right-bottom image shows the change of pixel values of projector input image for compensating the color distortion.

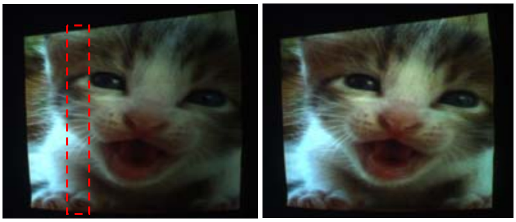


Fig. 10. Intensity-compensated projection. Left image: a projector lights the cylindrical surface on the left side and thus the left part of the image is brighter than the right part of the image. Right image: after compensation, the brightness of the whole image became similar.

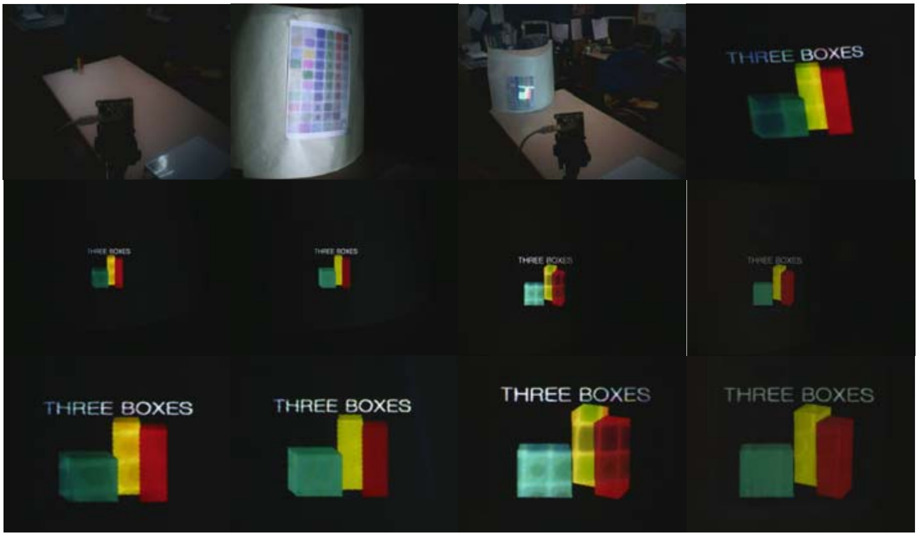


Fig. 11. Surface-independent direct-projected augmented reality. All the component methods are combined. First row: 3D target object (three blocks), nonplanar colored screen, without any compensation, magnification of the third image. Second row: after only geometric correction and after fully compensated projection when user's viewpoint is located on the left side of the screen, after only geometric correction and after fully compensated projection when user's viewpoint is located on the right side of the screen. Third row: magnification of the images of second row.

Figure 9 shows the result of compensating the color distortion using the radiometric compensation method. There is no color distortion in the left-bottom image of Fig. 9.

Figure 10 shows the result of intensity-compensated projection. The projector was calibrated and the geometry of the cylindrical surface was recovered. In the experiment, the brightness of the pixels within the red rectangle is the reference brightness and thus the brightness of the rest of the pixels was fit into the brightness.

The component methods for surface-independent DirectAR were combined to resolve mixed problems. The results are shown in Fig. 11.

4 Conclusion

A new framework for surface-independent DirectAR was presented. Various practical problems regarding DirectAR were addressed. Experimental results demonstrated that the problems unavoidable in realizing surface-independent DirectAR could be completely resolved using the proposed methods.

Currently, we are trying to develop an intelligent projection system using a single projector and apply to medical field. Developing an intelligent projection system using multiple projectors would be interesting as a future research.

Acknowledgement. This study was supported by a grant(02-PJ3-PG6-EV04-0003) of Ministry of Health and Welfare, Republic of Korea.

References

1. Grimson, W.E.L., etc.: An Automatic Registration Method for Frameless Stereotaxy, Image Guided Surgery, and Enhanced Reality Visualization. IEEE Transactions on Medical Imaging (1996)
2. Yasumuro, Y., Imura, M., Manabe, Y., Oshiro, O., Chihara, K.: Projection-Based Augmented Reality with Automated Shape Scanning. Proc. of SPIE EI (2005)
3. Snyder, J.M., Barr, A.H.: Ray Tracing Complex Models Containing Surface Tessellations. Proc. of SIGGRAPH, vol.21, no.4 (1987) 119-128
4. Grossberg, M.D., etc.: Making One Object Look Like Another: Controlling Appearance Using a Projector-Camera System. Proc. of CVPR (2004)
5. Hartley, R., Zisserman, A.: Multiple View Geometry. Cambridge University Press (2003)
6. Zhang, Z.: Flexible Camera Calibration by Viewing a Plane from Unknown Orientation. Proc. of ICCV (1999) 666-673
7. Surati, R.: Scalable Self-Calibrating Display Technology for Seamless Large-Scale Displays. PhD thesis, MIT (1999)
8. Raskar, R., etc.: iLamps: Geometrically Aware and Self-Configuring Projectors. Proc. of SIGGRAPH, vol.22 (2003) 809-818
9. NDI, http://www.ndigital.com/polaris.php
10. Mitsunaga T., etc.: Radiometric Self Calibration. Proc. of CVPR, vol.1 (1999) 374-380
11. Projector-related papers, http://www.cs.unc.edu/~raskar/Projector/projbib.html
12. Raskar, R., etc.: Shader Lamps: Animating Real Objects with Image-Based Illumination. Proc. of Eurographics Workshop on Rendering Techniques (2001) 89-102
13. Sukthankar, R., Stockton, R., Mullin, M.: Smarter Presentations: Exploiting Homography in Camera-Projector Systems. Proc. of ICCV (2001)
14. Bimber, O., Wetzstein, G., Emmerling, A., Nitschke, C.: Enabling View-Dependent Stereoscopic Projection in Real Environments. Proc. of ISMAR (2005) 14-23
15. Park, H., Lee, M.-H., Kim, S.-J., Park, J.-I.: Specular Reflection Elimination for Projection-Based Augmented Reality, Proc. of ISMAR (2005) 194-195

Aspects of Optimal Viewpoint Selection and Viewpoint Fusion

Frank Deinzer[1,⋆], Joachim Denzler[2], Christian Derichs[1,⋆], and Heinrich Niemann[1]

[1] Chair for Pattern Recognition, University Erlangen-Nürnberg,
91058 Erlangen, Germany
{deinzer, derichs, niemann}@informatik.uni-erlangen.de
[2] Chair for Image Processing, University Jena, 07737 Jena, Germany
denzler@informatik.uni-jena.de

Abstract. In the past decades, most object recognition systems were based on passive approaches. But in the last few years a lot of research was done in the field of active object recognition. In this context, there are several unique problems to be solved, such as the fusion of views and the selection of an optimal next viewpoint.

In this paper we present an approach to solve the problem of choosing optimal views (viewpoint selection) and the fusion of these for an optimal 3D object recognition (viewpoint fusion). We formally define the selection of additional views as an optimization problem and we show how to use reinforcement learning for viewpoint training and selection in continuous state spaces without user interaction. In this context we focus on the modeling of the reinforcement learning reward. We also present an approach for the fusion of multiple views based on density propagation, and discuss the advantages and disadvantages of two approaches for the practical evaluation of these densities, namely Parzen estimation and density trees.

1 Introduction

The results of 3D object classification and localization depend strongly on the images which have been taken of the object. For difficult data sets, usually more than one view is necessary to decide reliably on a certain object class. Viewpoint selection tackles exactly the problem of finding a sequence of optimal views to increase classification and localization results by avoiding ambiguous views or by sequentially ruling out possible object hypotheses. The optimality is not only defined with respect to the recognition rate, but also with respect to the number of views necessary to get reliable results.

In this paper, we present an approach for viewpoint selection based on reinforcement learning. The approach shows some major benefits: First, the optimal sequence of views is learned automatically in a training step without any user interaction. Second, the approach performs a fusion of the generated

⋆ This work was partially funded by DFG under grant SFB 603/TP B2. Only the authors are responsible for the content.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 902–912, 2006.

views, where the fusion method does not depend on a special classifier. This makes it applicable for a very wide range of applications. Third, the possible viewpoints are continuous, so that a discretization of the viewpoint space is avoided.

Viewpoint selection has been investigated in the past in several applications. Examples are 3D reconstruction [1] or optimal segmentation of image data [2]. In object recognition, some active approaches have already been discussed as well. [3] plans the next view for a movable camera based on probabilistic reasoning. The active part is the selection of a certain area of the image for feature selection. The selected part is also called the receptive field [4]. Compared to our approach, no camera movement is performed, neither during training nor during testing. Thus, the modeling of viewpoints in continuous 3D space is also avoided. The work of [5] uses Bayesian networks to decide on the next view to be taken. But the approach is limited to special recognition algorithms and to certain types of objects, for which the Bayesian network has been manually constructed. In other words, the approach is not classifier independent and cannot be applied without user interaction. [6] showed that the optimal action is the one that maximizes the mutual information between the observation and the state to be estimated.

In section 2 we will show how the viewpoint fusion of multiple views can be done based on recursive density propagation in a continuous state space. Our reinforcement learning approach for viewpoint selection is presented in section 3. The experimental results in section 4 show that the presented approach is able to learn an optimal strategy for viewpoint selection that generates only the minimal number of images. The paper concludes with a summary and an outlook to future work in section 5.

2 Viewpoint Fusion

In active object recognition, a series of observed images $\langle \boldsymbol{f} \rangle_t = \boldsymbol{f}_t, \boldsymbol{f}_{t-1}, \ldots, \boldsymbol{f}_0$ of an object are given together with the camera movements $\langle \boldsymbol{a} \rangle_{t-1} = \boldsymbol{a}_{t-1}, \ldots, \boldsymbol{a}_0$ between these images. Based on these observations of images and movements, one wants to draw conclusions for a non-observable state $\boldsymbol{q}_t$ of the object. This state $\boldsymbol{q}_t$ must contain both the *discrete* class Ω_κ and the *continuous* pose $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_J)^T$ of the object, leading to the state definition $\boldsymbol{q}_t = (\Omega_\kappa, \phi_1^t, \ldots, \phi_J^t)^T$. Please note that most related work is either restricted to just handling the class [7] or does not even claim to work on continuous poses [8]. The actions $\boldsymbol{a}_t$ consist of the *relative* camera movement with J degrees of freedom, in the following written as $\boldsymbol{a}_t = (\Delta\phi_1^t, \ldots, \Delta\phi_J^t)$. Generally, disturbances of these actions by some kind of inaccuracy within the movement have to be taken into consideration.

In the context of a Bayesian approach, the knowledge on the object's state is given in form of the a posteriori density $p(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1})$ and can be calculated from

$$p\left(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1}\right) = \frac{p\left(\boldsymbol{q}_t, \langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1}\right)}{p\left(\langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1}\right)} \tag{1}$$

$$= \frac{p\left(\boldsymbol{f}_t|\boldsymbol{q}_t, \langle\boldsymbol{f}\rangle_{t-1}, \langle\boldsymbol{a}\rangle_{t-1}\right) p\left(\boldsymbol{q}_t|\,\langle\boldsymbol{f}\rangle_{t-1}, \langle\boldsymbol{a}\rangle_{t-1}\right)}{\underbrace{p\left(\boldsymbol{f}_t|\,\langle\boldsymbol{f}\rangle_{t-1}, \langle\boldsymbol{a}\rangle_{t-1}\right)}_{=k}} \tag{2}$$

$$= \frac{1}{k} \cdot p\left(\boldsymbol{f}_t|\boldsymbol{q}_t\right) p\left(\boldsymbol{q}_t|\,\langle\boldsymbol{f}\rangle_{t-1}, \langle\boldsymbol{a}\rangle_{t-1}\right) \tag{3}$$

$$= \frac{1}{k} \cdot p\left(\boldsymbol{f}_t|\boldsymbol{q}_t\right) \int p\left(\boldsymbol{q}_t|\boldsymbol{q}_{t-1}, \boldsymbol{a}_{t-1}\right) \cdot p\left(\boldsymbol{q}_{t-1}|\,\langle\boldsymbol{f}\rangle_{t-1}, \langle\boldsymbol{a}\rangle_{t-2}\right) d\boldsymbol{q}_{t-1} \tag{4}$$

(1) results directly from the definition of the conditional probability $p(A|B) = p(AB)/p(B)$ and the further steps to (2) from the multiplication theorem for probability densities. In (3) the Markov assumption $p(\boldsymbol{f}_t|\boldsymbol{q}_t, \langle\boldsymbol{f}\rangle_{t-1}, \langle\boldsymbol{a}\rangle_{t-1}) = p(\boldsymbol{f}_t|\boldsymbol{q}_t)$ is applied. The formulation of $p(\boldsymbol{q}_t|\,\langle\boldsymbol{f}\rangle_{t-1}, \langle\boldsymbol{a}\rangle_{t-1})$ as an integral in (4) results from the total probability theorem. Obviously the probability (4) depends only on the camera movement $\boldsymbol{a}_{t-1}$. The inaccuracy of $\boldsymbol{a}_{t-1}$ is modeled within the state transition component $p(\boldsymbol{q}_t|\boldsymbol{q}_{t-1}, \boldsymbol{a}_{t-1})$.

The classic approach for solving this recursive density propagation is the Kalman Filter. But in computer vision, the necessary assumptions for the Kalman Filter ($p(\boldsymbol{f}_t|\boldsymbol{q}_t)$ being normally distributed) are often not valid due to object ambiguities, sensor noise, occlusion, etc. This is a problem since it leads to a distribution which is not analytically computable. An approach for the complicated handling of such multimodal densities are the so called particle filters [9]. The basic idea is to approximate the a posteriori density by a set of weighted samples. In our approach we use the Condensation Algorithm [9] which uses a sample set $Y_t = \{y_t^1, \ldots, y_t^K\}$ to approximate the multimodal probability distribution $p(\boldsymbol{q}_t|\,\langle\boldsymbol{f}\rangle_t, \langle\boldsymbol{a}\rangle_{t-1})$ by K samples $y_t^i = \{\boldsymbol{x}_t^i, p_t^i\}$. Each sample y consists of the position $\boldsymbol{x} = (\Omega_\kappa, \phi_1, \ldots, \phi_J)$ within the state space and a sample weighting p with $\sum_i p_t^i = 1$.

The Condensation Algorithm starts with an initial sample set Y_0. The samples of this set are distributed uniformly over the state space in our application as we have no knowledge given about the objects before observing the first image. For the generation of a new sample set Y_t, K new samples y_t^i are

1. drawn from Y_{t-1} with probability proportional to the sample weightings.
2. propagated with the necessarily predetermined sample transition model $\boldsymbol{x}_t^i = \boldsymbol{x}_{t-1}^i + (0, r_1, \ldots, r_J)^T$ with $r_j \sim \mathcal{N}(\Delta\phi_j^t, \sigma_j)$ and the variance parameters of the Gaussian transition noise σ_j.
3. evaluated in the image by $p(\boldsymbol{f}_t|\boldsymbol{x}_t^i)$. This evaluation is performed by the classifier. The only requirement for the classifier that shall be used together with our fusion approach is its ability to evaluate this density. In this work we use a classifier based on the continuous statistical eigenspace approach as presented in [10]. Other classifiers have been proven to work as well with the presented fusion approach.

In the context of our viewpoint selection, the densities which are represented by sample sets have to be evaluated. The direct evaluation of them beneath the positions given by the individual samples is not possible. It is necessary to find a continuous representation of the density. This will be done in two different ways in this paper.

Parzen estimation: A common way to evaluate non-parametric densities is the Parzen estimation [11] which is calculated from a sample set Y by

$$p(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t , \langle \boldsymbol{a} \rangle_{t-1}) \approx \frac{1}{K} \sum\nolimits_{i=1}^{K} g_0 \left(\boldsymbol{q}_t - \boldsymbol{x}_t^i \right) \quad , \tag{5}$$

with $g_0(\boldsymbol{v}) = \mathcal{N}(\boldsymbol{v}|\boldsymbol{\mu} = \boldsymbol{0}, \boldsymbol{\Sigma})$ denoting a windowing function. In this paper only a Gaussian window function is used. The choice of the mean vector $\boldsymbol{\mu} = \boldsymbol{0}$ is comprehensible as the difference $(\boldsymbol{q}_t - \boldsymbol{x}_t^i)$ in (5) results in zero-mean data. In contrast, the definition of the covariance matrix requires a careful consideration of methods like the mean minimal distance of samples or the entropy-based approach of [12] and will be omitted in this paper. For a more detailed explanation on the theoretical background of the approximation of (1) by a sample set we refer to [9].

Density trees: Another way to evaluate the densities represented by the sample set are the so-called *density trees* [13]. They use a tree structure to transform the discrete samples into a continuous density. Each node of the density tree represents a hyperrectangle in the state space over $\boldsymbol{q}$. A density is built by the repeated partitioning of the parameter space and refining the tree structure until a stop criterion is reached. A detailed description of that process is given in [13].

3 Viewpoint Selection

A straight forward and intuitive way to formalizing the problem is given by looking at Fig. 1. A closed loop between sensing s_t and acting $\boldsymbol{a}_t$ can be seen. The chosen *action* $\boldsymbol{a}_t$ corresponds to the executed camera movement, the sensed *state*

$$s_t = p(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t , \langle \boldsymbol{a} \rangle_{t-1}) \tag{6}$$

is the density as given in (1). Additionally, the classifier returns a so called *reward* r_t, which measures the quality of the chosen action resp. the resulting viewpoint. It is well known that the definition of the reward is an important aspect, as this reward should model the goal that has to be reached. Proper definitions for the reward in the context of our viewpoint selection problem are given later in this paper.

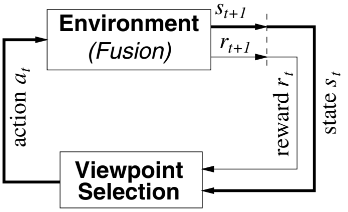


Fig. 1. Reinforcement learning

At time t during the decision process, i.e. the selection of a sequence of viewpoints, the goal will be to maximize the accumulated and weighted future rewards, called the *return*

$$R_t = \sum_{n=0}^{\infty} \gamma^n r_{t+n+1} \qquad \text{with } \gamma \in [0;1],\ 0^0 =: 1 \ . \tag{7}$$

The weight γ defines how much influence a future reward will have on the overall return R_t at time $t+n+1$. Of course, the future rewards cannot be observed at time step t. Thus, the following function, called the *action-value function* $Q\left(s, \boldsymbol{a}\right) = E\left\{R_t | s_t = s, \boldsymbol{a}_t = \boldsymbol{a}\right\}$ is defined, which describes the expected return when starting at an arbitrary time step t in state s with action $\boldsymbol{a}$. In other words, the function $Q\left(s, \boldsymbol{a}\right)$ models the expected quality of the chosen camera movement $\boldsymbol{a}$ for the future, if the viewpoint fusion has returned s before.

Viewpoint selection can now be defined as a two step approach: First, estimate the function $Q\left(s, \boldsymbol{a}\right)$ during training. Second, if at any time the viewpoint fusion returns s as classification result, select that camera movement which maximizes the expected accumulated and weighted rewards. This function is called the *policy*

$$\pi(s) = \underset{\boldsymbol{a}}{\operatorname{argmax}}\, Q\left(s, \boldsymbol{a}\right) \ . \tag{8}$$

The key issue of course is the estimation of the function $Q\left(s, \boldsymbol{a}\right)$, which is the basis for the decision process in (8). One of the demands defined in section 1 is that the selection of the most promising view should be learned without user interaction. Reinforcement learning provides many different algorithms to estimate the action value function based on a trial and error method [14]. Trial and error means that the system itself is responsible for trying certain actions in a certain state. The result of such a trial is then used to update $Q\left(\cdot, \cdot\right)$ and to improve its policy π.

As a result for the next episode one gets a new decision rule π_{k+1}, which is now computed by maximizing the updated action value function. This procedure is repeated until π_{k+1} converges to the optimal policy. The reader is referred to a detailed introduction to reinforcement learning [14] for a description of other ways for estimating the function $Q\left(\cdot, \cdot\right)$. Convergence proofs for several algorithms can be found in [15].

We are still missing the definition of the reward r_t. In the context of viewpoint selection the following two different definitions of rewards make sense.

Fixed Value: A way to model the goal is to define a reward that has a value of 0 except when reaching the terminal state:

$$r_{t+1} = \begin{cases} C & s_t \text{ is terminal state, } C > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

This approach has the advantage that the goal is defined very clearly. But the environment has to decide when the confidence of the classification is high enough to stop the viewpoint selection. If this decision is hard to make, no proper strategy will be learned. The advantage is that (9) maximizes the return of an episode for short episodes (at least for $\gamma \neq 0, \gamma \neq 1$). So this strategy promises to look for episodes with only a minimal number of views. In our work we use $C = 1.0$.

Entropy Based: Another approach follows the idea that viewpoints that increase the information observed so far should have large values for the reward. A well-known measure for expressing the informational content that fits our requirements is the negative entropy $-H$, yielding

$$r_{t+1} = -H(s_t) = -H\left(p(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t , \langle \boldsymbol{a} \rangle_{t-1})\right) . \tag{10}$$

In that sense the reward expresses the gain of knowledge about the object. (10) has the advantage that the goal is to improve the classification without only trying to reach a stop criterion. But it can not be made sure that maximizing the sum of entropies in (10) will always and under any circumstances lead to the absolutely shortest episodes.

Most of the algorithms in reinforcement learning treat the states and actions as discrete variables. Of course, in viewpoint selection parts of the state space (the pose of the object) and the action space (the camera movements) are continuous. A way to extend the algorithms to continuous reinforcement learning is to approximate the action-value function

$$\widehat{Q}\left(s, \boldsymbol{a}\right) = \frac{\sum\limits_{(s', \boldsymbol{a}')} K\left(d\left(\theta(s, \boldsymbol{a}), \theta(s', \boldsymbol{a}')\right)\right) Q\left(s', \boldsymbol{a}'\right)}{\sum\limits_{(s', \boldsymbol{a}')} K\left(d\left(\theta(s, \boldsymbol{a}), \theta(s', \boldsymbol{a}')\right)\right)} \tag{11}$$

which can be evaluated for any continuous state/action pair $(s, \boldsymbol{a})$. Basically, this is a weighted sum of the action-values $Q\left(s', \boldsymbol{a}'\right)$ of all previously collected state/action pairs $(s', \boldsymbol{a}')$. The other components within (11) are:

The **transformation function** $\theta(s, \boldsymbol{a})$ transforms a state s with a known action $\boldsymbol{a}$ with the intention of bringing a state to a "reference point" (required for the distance function in the next item). In the context of the current definition of the states from (6) it can be seen as a density transformation

$$\begin{aligned} \theta(s_t, \boldsymbol{a}_t) &= \theta\left(p(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t , \langle \boldsymbol{a} \rangle_{t-1}), \boldsymbol{a}_t\right) \\ &= \det\left(\boldsymbol{J}_{\boldsymbol{\zeta}_{\boldsymbol{a}_t}^{-1}}(\boldsymbol{q}_t)\right) p\left(\boldsymbol{\zeta}_{\boldsymbol{a}_t}^{-1}(\boldsymbol{q}_t) | \langle \boldsymbol{f} \rangle_t , \langle \boldsymbol{a} \rangle_{t-1})\right) \end{aligned} \tag{12}$$

with $\boldsymbol{\zeta}_{\boldsymbol{a}}^{-1}(\boldsymbol{q}) = (q_1 + a_1, \ldots, q_m + a_m)^T$. It has been shown in [16] that the density transformation simply performs a shift of the density, so that $\boldsymbol{J}_{\boldsymbol{\zeta}_{\boldsymbol{a}}^{-1}}(\boldsymbol{q}) = \boldsymbol{I}$.

A **distance function** $d(\cdot, \cdot)$ is needed to calculate the distance between two states. Generally speaking, similar states must result in low distances. The lower the distance, the more transferable the information from a learned action-value to the current situation is. As the transformation function (12) results in a density, the *Kullback-Leibler Distance* $d_{KL}(s_n, s'_m)$ between the two states $s_n = p(\boldsymbol{q} | \langle \boldsymbol{f} \rangle_n , \langle \boldsymbol{a} \rangle_{n-1})$ and $s'_m = p(\boldsymbol{q} | \langle \boldsymbol{f}' \rangle_m , \langle \boldsymbol{a}' \rangle_{m-1})$, which can easily be extended to a symmetric distance measure, the so called *extended Kullback-Leibler Distance* $d_{EKL}(s_n, s'_m) = d_{KL}(s_n, s'_m) + d_{KL}(s'_m, s_n,)$, can be used. Please note that in general there is no analytic solution for the extended Kullback-Leibler Distance, but as we represent our densities as sample sets anyway (see section 2), there are well-known ways to approximate it by Monte Carlo techniques. The Monte Carlo techniques will use either the Parzen estimation or the density trees to evaluate the densities.

A **kernel function** $K(\cdot)$ weights the calculated distances. A suitable kernel function is the Gaussian $K(x) = \exp(-x^2/D^2)$, where D denotes the width of the kernel. Low values for D will result in very detailed approximations provided that a lot of action-values $Q(s', \boldsymbol{a}')$ are available.

Viewpoint selection, i.e. the computation of the policy π, can now be written, according to (8), as an optimization problem which is solved in this work by applying a global Adaptive Random Search Algorithm [17] followed by a local Simplex:

$$\pi(s) = \underset{\boldsymbol{a}}{\operatorname{argmax}} \widehat{Q}(s, \boldsymbol{a}) . \tag{13}$$

4 Experimental Evaluation

Our primary goal in the experiments was to show that our approach is able to learn and perform an *optimal sequence* of views. We have shown in several publications (e.g. [18]) that the viewpoint fusion of a sequence of *randomly* chosen views works very well in real world environments and improves classification and localization result significantly. For that reason we decided to use the rather simple (from the pure object recognition's point of view) synthetic images of the two types of cups shown in Fig. 2 for the evaluation of our viewpoint selection approach. It was explicitly desired to have objects that can reach a 100% recognition rate given the optimal views.

The four cups of "type one" in the upper row of Fig. 2 show a number **1** or **2** on one, and a letter **A** or **B** on the other side. A differentiation between the 4 possible objects is only possible if number and letter have been observed and properly fused. The five cups of "type two" in the lower row of Fig. 2 show a number (**1 2 3 4 5**) on the front side. If this number is not visible the objects can not be distinguished or localized.

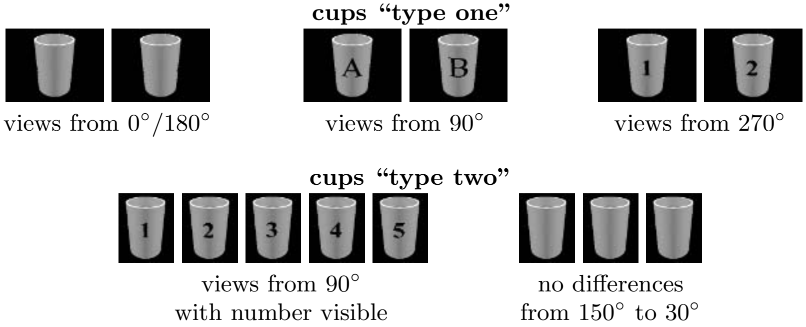


Fig. 2. Examples for objects that require viewpoint selection and fusion of images for proper recognition

The cups can be classified correctly and stably within an area of about 120°. Localization of the cups is possible within an area of approximately 140°. In our setup the camera is restricted, for both types of cups, to a movement around the object on a circle, so that the definition of the samples reduces to $\boldsymbol{x} = (\Omega_\kappa, \phi_1)$

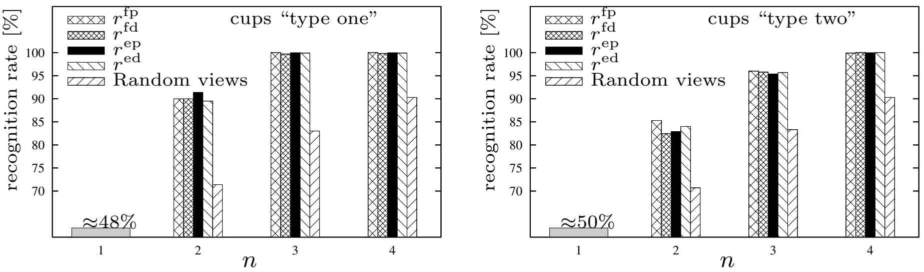


Fig. 3. Recognition rates of the viewpoint selection after planning n steps for the four different variations r^{ep}, r^{ed}, r^{fp}, r^{fd} compared to randomly chosen views. At step $n = 1$ all results are the same as no planning was done. These results compare to recognition rate that could be reached by pure passive recognition approaches. The parameters used for these results are $D = 50$ and $\gamma = 0.5$.

with actions $\boldsymbol{a}_t = (\Delta\phi_1^t)$, $\Delta\phi_1^t \in [0°, 360°]$. Our sample sets had size of $K = 1440$ (cups "type one") resp. $K = 1800$ (cups "type two") samples.

In our experiments we evaluated scenarios that differ in the way they evaluate the densities represented by our sample sets (Parzen estimation or density trees) and in the type of reward used (fixed value according to (9) or entropy-based as given in (10)), leaving the four variations r^{ep} (entropy-based reward, Parzen estimation), r^{ed} (entropy-based reward, density trees), r^{fp} (fixed value, Parzen estimation) and r^{fd} (fixed value, density trees). Additionally, three different values of the return parameter $\gamma \in \{0, 0.5, 1\}$ (see (7)) were used as they cover the two extreme values 0 and 1 which might have significant influence on the learned strategy and a value of 0.5 which represents the whole parameter range in-between. In a training step a total of 1000 episodes (with a maximal total length of 8 steps independent of the fact that the stop criterion was reached or not) were performed for every object, each value of $\gamma \in \{0, 0.5, 1\}$ and any of the variations r^{ep}, r^{ed}, r^{fp} and r^{fd}. The evaluation was performed on the results of a total of 1000 (for cups of "type one") resp. 1250 (for cups of "type two") episodes with randomly chosen classes and starting views.

In a first step we look at the recognition results of the viewpoint selection for the four variations r^{ep}, r^{ed}, r^{fp}, r^{fd} given values of $\gamma = 0.5$ and $D = 50$ in the kernel function $K(\cdot)$ for the approximation of the action-value function (11). As one can see in Fig. 3, the recognition results of the viewpoint selection reach a recognition rate of or close to 100%, as expected.

So the next question is if best viewpoints are selected in sense of the minimal numbers of views required. Number and letter are visible within the area stated above. Considering this, a theoretical minimum for the necessary mean sequence length exists:

- ≈ 2.2 views for the cups of "type one". Two views are required if number or letter is initially visible, three views otherwise.
- ≈ 2.0 for the cups of "type two". Depending on the strategy three to four views are required if number is not initially visible, one view otherwise.

Table 1. Mean number of views needed to allow for a reliable classification for different system settings. Object recognition stopped when the probability of the best class reached at least 95%.

cups "type one"			
Variation	D=2, γ=...		
	0	0.5	1
r^{fp}	2.17	2.18	2.40
r^{fd}	2.28	2.29	5.70
r^{ep}	2.19	2.17	2.20
r^{ed}	2.21	2.23	2.22

cups "type two"			
Variation	D=2, γ=...		
	0	0.5	1
r^{fp}	2.06	2.00	2.15
r^{fd}	2.02	2.03	2.82
r^{ep}	2.06	2.01	2.05
r^{ed}	2.10	2.02	2.00

> Anyhow, the theoretical minimum for the necessary mean sequence length can be shown to be always ≈ 2.0 steps.

Setting $D = 2$ since 1000 training episodes justify a detailed approximation and stopping when the probability of the best class is at least 95%, the mean number of views required to reach the stop criterion are summarized in Table 1. These numbers show that most configurations are very close to the theoretical minimum of required views. Exceptions are the variations r^{fp} and r^{fd} in combination with $\gamma = 1.0$. The reason can be found in the definition of the this reward in (9). Above we mentioned that the reward has to model the intended goal. This was done correctly in (9) but a reward of 0 means that if the end of the episode is not reached with the next step no "costs" are caused. In combination with $\gamma = 1$ this results in a total return according to (7) that is 1 independent of the length of the episode. In the sense of reinforcement learning, there is no need for the agent to look for short episodes. As one can see by means of the rightmost graph of the approximated action-value function in Fig. 4 no proper strategy was learned since all possible actions show nearly the same value. The small dents at 0° and 180° result from the limitations of the episode length to 8 steps (see above) that forces the system to learn at least a little bit of knowledge. This behavior could be changed in (9) if a negative value instead of 0 is returned. This could be seen as costs that force the agent to minimize the episode length. But the discussion of how to properly model costs in viewpoint selection is outside the scope of this paper.

Another observation from Table 1 is that the results for the variations that use the Parzen estimation for the evaluation of the densities $p(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1})$ are better than the ones that use the density trees. The reason is obvious if one looks at the left and middle approximated action-value function in Fig. 4. The variations that use the Parzen estimation have a smoothly approximated action-value function. In contrast the approximations of the density trees are highly jagged. This is due to the nature of the density trees: They approximate densities by piecewise constant values, leading to densities that are not continuous.

The computational effort and the required memory resources for planning a new viewpoint is rather high. For the cups of "type one" one planning step, i.e. the evaluation of (13), requires 550 to 750 evaluations of (11), each lasting

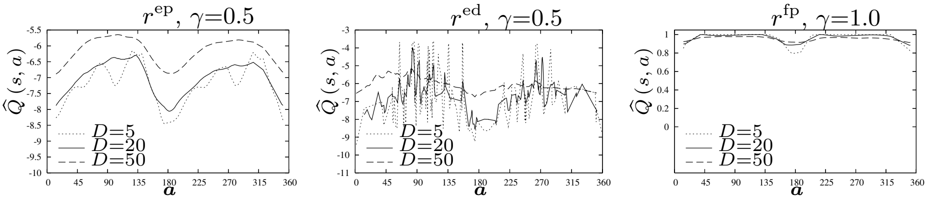


Fig. 4. Influence of D, γ and the type of reward and density evaluation to the approximation of the action-value function. All graphs show the estimated quality of the possible action for a current view of 0° to the cups of "type one". Graphs for $\gamma = 0$ are very similar to the ones with $\gamma = 0.5$ and omitted for that reason.

≈130ms. The 3670 action-values collected during the 1000 training episodes allocate 371 MB of memory if using the Parzen estimation and 96 MB for the density trees. The cups of "type two" require between 120 and 220 evaluations of (11) each lasting ≈150ms. The memory allocation for storing the 3160 action-values of the 1000 training episodes is 382 MB (Parzen estimation) resp. 116 MB (density trees). All numbers were evaluated on a Linux PC with a Xeon 2.80 GHz processor and 2 GB of main memory.

The conclusion of the experiments are that both types of introduced rewards lead to good planning results, at least for $\gamma \neq 1$. The necessary evaluation of the densities from the viewpoint fusion should be done with the Parzen estimation although the results of the density trees are better than the approximated action-value functions promise. If lack of memory is a problem the density tree variations might be an interesting alternative as they show huge memory saving compared to the Parzen estimation.

5 Summary and Future Work

In this paper we have presented the impact of several types of rewards and approaches for working with the densities given by the viewpoint fusion on the recognition rates of our general framework for viewpoint selection. We discussed several aspects of how to model the reward and the effects of different approaches for the evaluation of densities given as sample sets by the viewpoint fusion.

The viewpoint selection works in continuous state and action spaces and is independent of the chosen statistical classifier. Furthermore, the system can be trained automatically without user interaction. The experimental results on two objects that require different strategies for recognition have shown that an optimal planning strategy was learned.

In our future work we will evaluate how much the planning of optimal view sequences improves object recognition rates on real world objects compared to the random strategy we used in [18]. Finally, for higher dimensional state spaces, other reinforcement learning methods might be necessary to reduce training complexity.

References

1. P. Lehel and E.E. Hemayed and A.A. Farag: Sensor Planning for a Trinocular Active Vision System. In: CVPR. (1999) II:306–312
2. Madsen, C., Christensen, H.: A Viewpoint Planning Strategy for Determining True Angles on Polyhedral Objects by Camera Alignment. PAMI **19** (1997)
3. Roy, S.D., Chaudhury, S., Banerjee, S.: Recognizing Large 3-D Objects through Next View Planning using an Uncalibrated Camera. In: ICCV 2001, Vancouver, Canada (2001) II: 276 – 281
4. Schiele, B., Crowley, J.: Transinformation for Active Object Recognition. In: ICCV 98, Bombay, India (1998) 249–254
5. Krebs, B., Burkhardt, M., Korn, B.: Handling Uncertainty in 3D Object Recognition using Bayesian Networks. In: ECCV 98, Berlin (1998) 782–795
6. Denzler, J., Brown, C.: Information theoretic sensor data selection for active object recognition and state estimation. PAMI **24** (2002)
7. Zhou, X., Comaniciu, D., Krishnan, A.: Conditional feature sensitivity: A unifying view on active recognition and feature selection. In: ICCV 03, Nice, France (2003)
8. Callari, G., P.Ferrie, F.: Active Object Recognition: Looking for Differences. International Journal of Computer Vision **43** (2001) 189–204
9. Isard, M., Andrew, B.: CONDENSATION — Conditional Density Propagation for Visual Tracking. IJCV 98 **29** (1998) 5–28
10. Gräßl, C., Deinzer, F., Mattern, F., Niemann, H.: Improving Statistical Object Recognition Approaches by a Parameterization of Normal Distributions. Pattern Recognition and Image Analysis **14** (2004) 222–230
11. Parzen, E.: On the estimation of a probability density function and mode. Annals of Mathematical Statistics **33** (1962) 1065–1076
12. Viola, P.: Alignment by Maximization of Mutual Information. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, Cambridge, Massachusetts (1995)
13. Thrun, S., Langford, J.: Monte carlo hidden Markov models. Technical Report CMU-CS-98-179, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA (1998)
14. Sutton, R., Barto, A.: Reinforcement Learning. A Bradford Book, Cambridge, London (1998)
15. Bertsekas, D., Tstsiklis, J.: Neuro–Dynamic Programming. Athena Scientific (1996)
16. Deinzer, F., Denzler, J., Niemann, H.: Viewpoint Selection – Planning Optimal Sequences of Views for Object Recognition. In: Computer Analysis of Images and Patterns. Volume 2756 of LNCS., Springer (2003) 65–73
17. Törn, A., Žilinskas, A.: Global Optimization. Volume 350 of Lecture Notes in Computer Science. Springer, Heidelberg (1987)
18. Deinzer, F., Denzler, J., Niemann, H.: On Fusion of Multiple Views for Active Object Recognition. In: DAGM 2001, Berlin, Springer (2001) 239–245

An Efficient Approach for Multi-view Face Animation Based on Quasi 3D Model

Yanghua Liu, Guangyou Xu, and Linmi Tao

Key Lab. on Pervasive Computing, Ministry of Education,
Tsinghua Univ., Beijing, China
liuyangh00@mails.tsinghua.edu.cn

Abstract. Multi-view face animation is widely required in various applications nowadays, but most of existing relative techniques have burdensome computational load in data registration, model construction and animation. Addressing those problems, this paper proposes an efficient approach via employing quasi 3D model, which is the fusion of a 3D geometry model with 2D facial textures. 3D *Point Distribute Model* (PDM) serves to model geometrical deformation in the shape model. By preserving depth information, the proposed approach is convenient to manipulate pose variation and deal with new-coming subjects. By taking advantages of image-based techniques, data collection is facile and the number of our 3D model vertices is reduced to less than one hundred, while animated faces still keep expressive by incorporating with partial *Expression Ratio Image* (ERI). The primary experiments demonstrate that our approach efficiently achieved individual animated face among viewpoint range of [-60, 60] based on only 2 input facial images.

1 Introduction

Multi-view face animation is widely required in video conference, visual presidents, SMS reader for mobile-phone, and other applications. Real-time and easiness for people to use are two important demands for face animations in those applications. However most of portable devices are with relatively low computing power, thus efficient vivid multi-view face animation, which needs to achieve both video-realistic facial textures and natural head movement, is a significant challenge. And it is also much trouble to animate different individuals' faces without a lot of repeated work as data registration and training. Existing face animation techniques are respectively in 3D [6], [9], [11], [13] or 2D [1], [2], [3], [8].

Face animation exhibits both facial texture changes and facial motion. In 3D model-based animation approaches, 3D facial model integrating 3D shape and 3D texture is built up to parameterize facial geometry and texture for each individual, and then manipulate the surfaces of face over time [9], [11]. Taking advantages of depth information, 3D model-based animation approaches are competent in pose variability and occlusion estimation. Concerning facial texture, intricate diversification of facial textures root in intricate motions of human face, such as mouth self-occlusion caused by motion of lips and teeth, subtle wrinkles around mouth corners caused by smiling, and deep furrows on facial forehead caused by frowning. To obtain 3D facial texture

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 913 – 922, 2006.

for animation, one adoptable way is to use laser scanners, but it is too expensive and rare for common people. In other techniques, photogrammetric techniques, dealing with shading, edge, facial features and so on, are required to retrieve precise 3D information from images or video [11]. Pighin [11] captured 3D face geometry and textures by correspondences in a set of calibrated images, but they may be too difficult to achieve vast expressional 3D facial textures so that naturalness of animation is limited. 3D model-based techniques do have difficulties in capturing dynamic 3D facial textures during data collection and registration. Additionally, 3D facial models usually have thousands of vertices, and improvement of animation reality for those models may cost much animation complexity and rendering latency.

Compared with 3D approaches, image-based animation approaches are superior in visual effect [8]. By analyzing and training on a set of collecting images, image textures' information is organized and structured serving for synthesis of animated facial texture [1], [8]. Collection of 2D facial images is much tractable than obtaining of 3D facial textures. Ezzat [8] employed *Multidimensional Morphable Model* (MMM) and achieved novel pronouncing video with intricate mouth texture from a small number of prototype images, but the system can only generate pose-fixed facial animation. Different poses bring much non-linearity in both shape and texture. In 2D, due to lack of depth information, pose variations and self-occlusions are too difficult to be treated with, though there are some view-based approaches. Besides, it is also inconvenient to animate a new-coming face without training, so they are not suitable for modeling different individuals' face animation.

For texture synthesis, morph-based approaches and geometric-warping-based approaches are the two basic classes of techniques. Those techniques are based on interpolation or 2D transfiguration [1], [7], [8], [11], but they can only capture facial geometrical changes and 2D transfiguration, completely ignoring expression details. The reason is that those approaches have no sense on depth variation; so that the 3D objects' appearance affected by illumination is disregarded. Moreover, they take no consideration into subtle detail changes in illumination and appearance, such as wrinkles near canthus, crease in forehead, which are actually expressive and maybe important visual cues [14], [15]. To represent expression details, simple facial shape approximation by 2D texture warping should be improved. *Expression Ration Image* (ERI) was introduced by Liu et al [14]. ERI can capture and generate detailed texture diversification caused by illumination changes, hence makes the animated face more expressive and convincing. It is very helpful for vivid face animation [12].

Both 3D and 2D techniques have their own intrinsical problems. This inspires us that the advantages of two kind techniques should be fused to improve performance of animation and they can supply a gap for each other.

For the goal of different users' vivid multi-view face animation with low system requirement, we propose an approach based on our quasi 3D face model which is a fusion of 3D techniques and 2D techniques. In our previous work, based on quasi 3D model, we have achieved a given subject's real-time multi-view face animation by training on a subject's two synchronously captured corpuses [7]. But for each given subject, a training and analysis step is inescapable. So in this work for convenient personalized face animation, the main problems are how to conveniently personalize quasi 3D model and how to apply deformation to new-coming model.

Quasi 3D face model is constituted by shape model and texture model. Considering that depth information gives much advantage in dealing with facial pose variety and self-occlusion estimation, essential 3D information is preserved in 3D facial geometry model. *Point Distribute Model* (PDM) [4] serves to model geometrical deformation.

The correspondence of 3D meshes and 2D meshes serves as the cues between shape and texture model in quasi 3D facial model. The 2D meshes of texture model are rightly the projection of 3D meshes of the shape models, so in this way the texture model is well integrated with the shape model. 2D texture space incorporated with partial *Expression Ration Image* (ERI) is employed as the texture model. ERI is trained based on a collection of example images from a corpus and applied in part of facial area where movements of facial muscles are complicated.

The remainder of this paper is organized as follows. The framework of our approach based on quasi 3D face model is introduced in section 2. In section 3, the shape model is described, including individualization of generic 3D facial geometry model and the adjustment of vertex motion vectors. Section 4 introduces construction of partial ERI, synthesis of animated face and pro-processing. The experiment results are shown in section 5 and conclusions are drawn in section 6.

2 Framework of Our Approach

The framework of our approach based on quasi 3D model is shown in Fig. 1. Quasi 3D model is composed of a shape model and a texture model. The shape model contains a 3D facial geometry model and vertex motion vectors. Vertex motion vectors are vectors representing the moving direction and magnitude for each vertex and in form of sparse 3D PDM (*Point Distributed Model*) [7]. It flexibly represents allowed geometry variability for face deformation and preserves important depth information for pose and occlusion estimation. To attain an individual's 3D facial geometry model

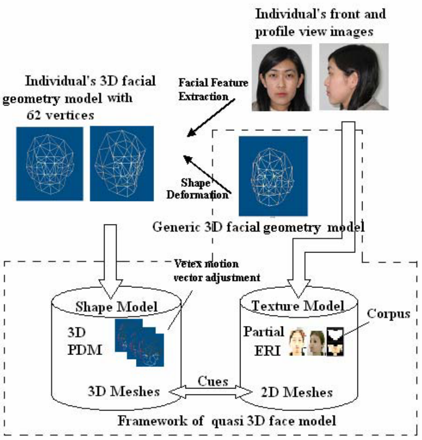

Individual's front and profile view images
Individual's 3D facial geometry model with 62 vertices
Facial Feature Extraction
Shape Deformation
Generic 3D facial geometry model
Vetex motion vector adjustment
Shape Model
3D PDM
3D Meshes
Cues
Texture Model
Partial ERI
Corpus
2D Meshes
Framework of quasi 3D face model

Fig. 1. Framework of quasi 3D model. The model is composed of a shape model, which contains a 3D facial geometry model and vertex motion vectors in form of PDM, and a texture model, which is a 2D texture space integrated with partial ERI.

conveniently, a generic 3D facial geometry model is individualized by two correspondent face images. Furthermore, for animation, the vertex motion vectors, should be adjusted too. Those vectors are adjusted according to the scale ratio between personalized and generic model. By this way, personalized facial animation shape can be achieved by applying personalized vertex animation vector on personalized model.

The texture model is facial texture space integrated with partial *Expression Ratio Image* (ERI). It describes both texture deformations caused by facial features' displacements and detailed texture diversifications caused by illumination changes. The shape and texture model is connected by correspondence of 3D and 2D meshes. Because 2D texture space has strong capability in representing natural appearance, the density of vertices in 3D geometry model can be comparatively sparse. The facial geometry model we employed has only 62 vertices.

We expressed quasi 3D model with a parameter set as follow:

$$C = (P, V, T, \alpha, \beta, \gamma)^T \tag{1}$$

where P denotes vertex set of 3D facial geometry model as $\{p_i\}_{i=1}^{N}$ (*N* is the number of vertices), V denotes vertex motion vectors as $\{v_i\}_{i=1}^{N}$, and α, β, γ respectively represents Euler angles of face motion. T denotes facial textures as $\{T_i\}_{i=1}^{L}$, and it is the combination of partial ERI texture $\{T_i^E\}_{i=1}^{L}$ (L is the number of prototype images) and non-ERI texture $\{T_i^{NE}\}_{i=1}^{L}$.

3 Shape Model of Quasi 3D Model

To construct shape model of quasi 3D model, firstly, individual's facial geometry model is achieved with aid of a generic 3D geometry model. Then generic vertex motion vectors in PDM, which have been achieved in our previous work [7], should be adjusted according to the individual's geometry model.

3.1 Generation of Personalized 3D Geometry Model

To generate 3D geometry model for individual is to achieve geometrical positions for each 3D vertex. Firstly from the inputted front-view and profile-view facial images, fiducial facial features (32 for front-view and 16 for profile-view) are extracted by view-based ASMs [16]. Then the generic 3D facial geometry model is deformed according to the facial features by *Radial Basis Function* (RBF).

RBF is an approach to achieve interpolation between two corresponding datasets [5]. Supposing that all positions of the first dataset and key points of the second data set are known, the position of other points of the second dataset can be interpolated by the corresponding relationship of the known key points' pairs. Given two corresponding 3D datasets $P = \{p_i\}_{i=1}^{N}$ and $P^D = \{p_i^D\}_{i=1}^{N}$, an arbitrary subset $Q = \{q_j\}_{j=1}^{M}$ (*N* and *M* is respectively the number of points in dataset *P* and *Q*.) and a Radial Basis Function $\phi_i(r)$ (r is the supporting radius which is used to control the density of interpolation), the deformation can be determined by equation (2).

$$
\begin{aligned}
&d(Q) = & \quad & d(p_i) = p_i^D \,|_{i=1}^N \\
&c_0 + [c_1 \quad c_2 \quad c_3]Q & & \sum_{i=1}^{N} \lambda_i = 0 \\
&+\sum_{i=1}^{N} \lambda_i \phi_i(|Q-P|) & & \sum_{i=1}^{N} p_{i,j}\lambda_i = 0\,|_{j=x,y,z}
\end{aligned}
\qquad (2)
$$

where c_0 , c_1, c_2 , c_3 and λ_i are coefficients determined by equation (2).

The deformation of the generic 3D facial geometry model is based on assumption of orthographic between front and profile, and performed in three steps (see Fig. 2).

1) Given known facial feature points in front-view image and the generic model, apply RBF based deformation and interpolation in X and Y coordinates, and set the value of each vertex's position in Z coordinate the same as the generic model.
2) Given known facial feature points in profile-view image and the generic model, apply RBF based deformation and interpolation in Y and Z dimension and keep the X coordinate values.
3) Given all the feature points in front and profile view images, apply RBF based deformation and interpolation in all dimension.

After these three RBF deformations, an adapted individual's 3D face geometry model is easily obtained with only 2 inputted facial images.

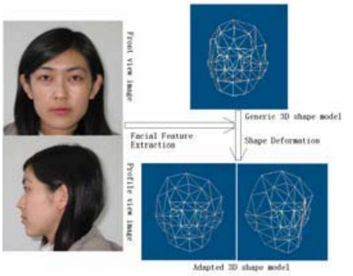


Fig. 2. Generation of 3D facial geometry model based on a generic 3D facial geometry model and two facial images (front and profile view). 32 facial feature points are extracted from front-view image and 16 from profile-view. Then 3-step RBF deformation is performed.

3.2 Vertex Motion Vector Adjustment

3D generic PDM has been trained in our previous work [7], so the allowed deformations of animation have been preserved in the PDM. The generic vertex motion vectors for each vertex can be decoded from PDM [7]. Those vectors should be adjusted to adapt to the individual's geometry model so that can be used to drive the correspondent geometry model. Because in the deformed facial geometry model, the relative positions and relative distances of vertices may have been changed, both direction and magnitude of the motion vectors should be adjusted [10].

Direction adjustment is carried out firstly. As the topology between the generic and deformed individual's 3D geometry model is consistent, the direction adjustment of vertex motion vectors can be performed for each pair of correspondent vertices. The adjustment is based on local coordination system transformation. Local coordination system for each vertex in geometry model is constructed as following.

For each vertex, among the meshes sharing this vertex, the X-axis is the average of meshes' normal. Given plane ϕ is the vertical plane of X-axis and the Y-axis is the projection of any connected edge onto ϕ, then Z-axis is the cross product of X and Y. Supposing that matrix ${}^{S}_{W}R$ denotes the rotation from a local source vertex coordinate to the world coordinate, ${}^{W}_{D}R$ from the world coordinate to the local deformed vertex coordinate, and ${}^{S}_{D}R$ from a local source vertex coordinate to the local deformed vertex coordinate, ${}^{S}_{D}R$ can be calculated as equation (3).

$$ {}^{S}_{D}R={}^{W}_{D}R\,{}^{S}_{W}R \qquad v^{T}_{i,a}={}^{S}_{D}R_{v_i}v_{i,a}={}^{W}_{D}R_{v_i}\,{}^{S}_{W}R_{v_i}v_{i,a} \tag{3} $$

Given ${}^{S}_{W}R_{v_i}$, ${}^{W}_{D}R_{v_i}$ and a vertex motion vector $v_{i,a}$ which is applied on vertex v_i, transformed correspondent vertex motion vector $v^{T}_{i,a}$ can be computed as equation (3).

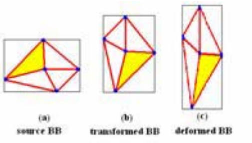


Fig. 3. Local Bounding Box for vertex motion vector magnitude adjustment. (a) is the source BB, (b) is the transformed BB by multiplying ${}^{S}_{D}R$, and (c) is the deformed BB. The local scale factor θ is decided by the proportion of size (c) and (b).

Secondly, magnitude adjustment of vertex motion vector for each vertex is performed. The magnitude adjustment is decided by local scale factor θ, which is the proportion of the local scales at correspondent vertices. It should be noticed that θ is restrained by a global threshold, in case that there may be too large geometrical difference between the source model and deformed model., Local scale of each vertex is defined by local Bounding Box (BB) around the meshes sharing the vertex as Fig. 3 shown. To get a fair comparison of local scale and eliminate the disturbance of rotation, the source BB B_S is transformed to B_T by multiplying rotation matrix ${}^{S}_{D}R$. Given deformed BB B_D , θ is computed as equation (4). Then motion vector adjustment factor A_{V_i} for each vertex v_i is equal to $\theta_{v_i}{}^{S}_{D}R_{v_i}$, and the deformed animation vector $v^{D}_{i,a}$ can be calculated as equation (4).

$$ \theta=\frac{Size_{B_D}\,|_{x,y,z}}{Size_{B_T}\,|_{x,y,z}} \qquad v^{D}_{i,a}=A_{v_i}v_{i,a}=\theta_{v_i}{}^{S}_{D}R_{v_i}v_{i,a}\,. \tag{4} $$

4 Texture Model for Quasi 3D Model

The texture model of quasi 3D model is not 3D facial texture model but 2D facial texture space integrated with partial ERI. Two facial textures respectively from front-view and profile-view have been captured for each individual. ERI is employed in the facial area with abundant expressions. 2D meshes are aligned on the front-view images and they are rightly the projection meshes of 3D geometry meshes (Fig. 4).

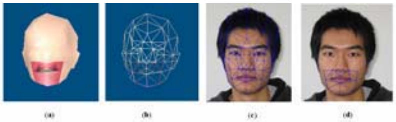


Fig. 4. Meshes and images for texture model. (a) is 3D meshes; (b) is 2D meshes; (c) is 2D meshes with texture and (d) is 2D meshes for partial ERI.

4.1 Training of Partial ERI

Partial ERI can enhance facial expression mapping with illumination changes [14]. Based on a collected corpus data [7], partial ERI is trained for each phoneme in mouth and its neighborhood area (see in Fig. 4(d)).

Key-frames $\{I_i\}_{i=1}^{L}$ are achieved by *Principal Component Analysis* (PCA) (Please refer to [7]). A reference image I_1 without any expressions and motion (named NA), is defined in image set $\{I_i\}_{i=1}^{L}$. All the other key-frame images $\{I_i\}_{i=2}^{L}$ are aligned to reference image I_1 sharing same shape with I_1. Then the ratio of expression image can be calculated by equation (5).

$$R_i(u,v)\,|_{i=2}^{L} = I_i(u,v)\,/\,I_1(u,v)\,|_{i=2}^{L} \tag{5}$$

where (u,v) denotes the coordinates of a pixel in the images, and $I_i(u,v)$ denotes the color values of the pixel in image I_i. Given a new individual's NA facial texture T_1, the correspondent expression texture T_i^E can be computed by equation (6).

$$T_i^E(u,v)\,|_{i=2}^{L} = T_1^E(u,v) \times R_i(u,v)\,|_{i=2}^{L}\,. \tag{6}$$

4.2 Synthesis and Pro-processing

There are four steps in animation synthesis: shape generation, texture generation, trajectory synthesis and pro-processing. Since $V = \{v_i\}_{i=1}^{N}$ in 3D PDM have been achieved by model fitting and trajectory distribution parameters has been trained in our previous work, the new individual's vertex motion vectors $V_a^D = \{v_{i,a}^D\}_{i=1}^{N}$ can also be acquired by adjustment in direction and magnitude (see section 3.2). When V_a^D is applied to the individual's geometry model $P^D = \{p_i^D\}_{i=1}^N$ (section 3.1), animated 3D geometry P_a^D is generated. According to α, β, γ, 2D animated shape is attained by projecting 3D geometry P_a^D followed with occlusion estimation.

The synthesis of texture is to apply partial ERI to achieve T_i^E, and then warp T_i^E and T_i^n to the generated 2D shape according to the 2D meshes.

Trajectory synthesis is processed to make animation imaged sequence smoothly as we have done in [7]. During pro-processing, boundary of animated face is smoothed and blurred. It must be noticed that individual's inside mouth texture including teeth and tongue texture is not captured in data collection. So the generic inside-mouth texture is adopted when mouth is open.

5 Experiment Result

We focused on facial expressions in mouth area, where movements of facial features and diversifications of facial textures are most abundant, to experiment face animation based on quasi 3D model.

In shape modeling, individuals' pair images from front and profile view were captured and used to personalize generic 3D geometry model (section 3.1). Our 3D geometry model has only 62 vertices and 102 meshes, among which there are 14

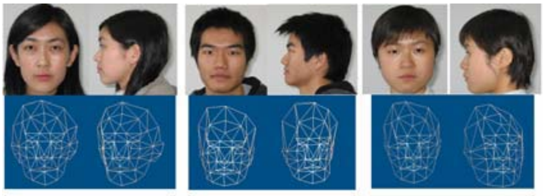


Fig. 5. Facial images (front and profile view) and the correspondent personalized 3D geometry models (front and half-profile view)

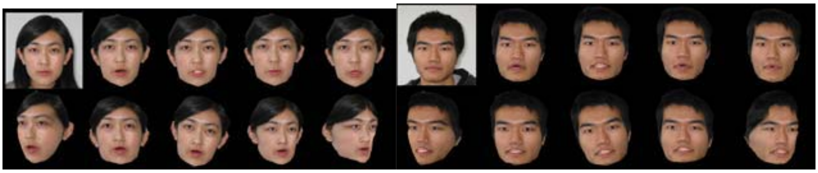


Fig. 6. Different individuals' talking faces selected from their generated talking video. The first image of each person is the input front view image, the male.The first row: a female talking head pronouncing different phoneme (/ang/, /q/, /m/, /y/ in Chinese) with same pose; the second row: the female talking head pronouncing a same phoneme with different poses.the third row: a male talking head pronouncing different phoneme (/ang/, /q/, /g/, /u/ in Chinese) with same pose; the fourth row: the male pronouncing a same phoneme with different poses.

meshes for partial ERI around mouth area (Fig. 5). After vertex motion vectors personalization, personalized vertex motion vectors were used to generate personalized animation shape. The shape variation is still represented in a low dimension in virtue of PDM.

In Fig. 6, two different individuals' talking faces are shown. The time for creating an animation image takes about 100ms in Pentium 4-M CPU 2.00GHz processor. We only used front-view facial texture for animation synthesis, so the experiment viewpoint range is [-60, 60]. If profile-view texture is also used, the possible view range can be as wide as [-90, 90].

6 Conclusion

We presented a novel approach for realistic multi-view face animation based on quasi 3D model. With only two face images, front and profile view respectively, different individual's face animation allowing head movement is generated without any other training. And the animation is achieved with low computational requirement in virtue of PDM. Fusing Advantages of 3D and 2D animation techniques, the limitations of both two techniques are weakened. Mouth area has the most complex expressions, our approach captured the details of mouth motion, which can be extended to model the other face expression. Considering low computational and resolving power of mobile devices, we reduced the number of geometry model vertices to 62, which is the minimum for realistic animation based on current experiments. The geometrical model can be designed and adjusted according to the computational power of mobile devices, also the demand of animation naturalness. Our experiments on pro-processing suggested that the density of vertices along facial contour plays key role in generating natural animation.

Presently, the animations of all individuals are performed in the same style without their own acting characteristic. How to learn a personal acting style easily is our research work in future.

Acknowledgment

This work is supported by Project 60433030 of NSFC.

References

1. T. Ezzat and T. Poggio, Miketalk: a talking facial display based on morphing visemes, In Proceedings of the Computer Animation Conference (1998).
2. S. Romdhani, S. Gong, and A. Psarrou, “A Multi-view Nonlinear Active Shape Model using Kernel PCA”, In British Machine Vision Conference, Nottingham, UK, (1999) 483–492
3. T. Cootes, G. Edwards, and C. Taylor, Active Appearance Models, In European Conference on Computer Vision, volume 2, Freiburg, Germany, (1998), 484–498
4. Cootes TF, Taylor CJ, Cooper DH, Graham J. Active Shape Models - Their Training and Application, Computer Vision and Image Understanding, 61(1), (1995) 38-59

5. Hui Zhang, Vladimir Vezhnevets, and Heui-Keun Choh, Image-based Photorealistic 3D Face Modeling, In Sixth IEEE International Conference on Automatic face and Gesture Recognition, Seoul, Korea, May, (2004), 49-56
6. F.I. Parke. Computer generated animation of faces, Proceedings ACM annual conference, August (1972)
7. Yanghua Liu, Guangyou Xu, and Qiang Wang. Realistic Multi-view Face Animation with aid of 3D PDM, In Sixth IEEE International Conference on Automatic face and Gesture Recognition, Seoul, Korea, May, (2004) 511-518
8. T. Ezzat, G. Geiger and T.Poggio, Trainable Videorealistic Speech Animation, In Proceedings of ACM SIGGRAPH 2002, San Antonio, Texas, July (2002) 388-398
9. D. Terzopoulos and K. Waters, Physically-based facial modeling, analysis, and animation, J. of Visualization and Computer Animation, vol. 1 March, (1990) 73-80
10. Jun-yong Noh, Ulrich Neumann, Expression Cloning, SIGGRAPH 2001, Los Angeles, August (2001) 277-288
11. F. Pighin, Modeling and Animating Realistic Faces from Images, A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, (1999)
12. Wenhui Zhu, Yiqiang Chen, Yanfeng Sun, Baocai Yin and Dalong Jiang, SVR-Based Facial Texture Driving for Realistic Expression Synthesis, In Third International Conference on Image and Graphics, Hong Kong, China, (2004) 456-459
13. Y. Lee, D. Terzopoulos, and K. Waters, Realistic Modeling for Facial Animation, In Proceedings of SIGGRAPH 1995, ACM Press / ACM SIGGRAPH, Annual Conference Series, ACM, Los Angeles, California, (1995) 55-62
14. Zicheng Liu, Ying Shan, Zhengyou Zhang. Expressive Expression Mapping with Ratio Images, SIGGRAPH 2001, Los Angeles, August (2001) 271-276
15. Jiang DaLong, Gao Wen, Wang ZhaoQi, Chen YiQiang, Realistic 3D Facial Animations with Partial Expression, Chinese Journal of Computers, Vol. 27, No. 6, (2004) 750-757
16. Yanghua Liu,Yang Li, Linmi Tao, Guangyou Xu, Multi-view Face Alignment Guided by Several Facial Feature Points, In Proceedings of the Third International Conference on Image and Graphics (2004) 238-241.

Hallucinating 3D Faces*

Shiqi Peng, Gang Pan**, Shi Han, and Yueming Wang

Dept. of Computer Science, Zhejiang University, Hangzhou 310027, China
{apex, gpan}@zju.edu.cn

Abstract. Super resolution technique could produce a higher resolution image than the originally captured ones. However, a few works have been done in 3D models. In this paper, we focus on generating the 3D face model of higher resolution from one input of 3D face model. In our method, the 3D face models including the training samples and input model are remeshed to construct consistent mesh to find out the correspondence among the models. The super resolution then is performed in the remeshed models to reconstruct the high resolution model. The experiments using USF HumanID 3D face database of 100 3D face models are carried out, and demonstrate the presented algorithm is promising.

1 Introduction

Super-resolution is a technique that could obtain the higher resolution image from the originally captured ones. Numerous approaches have been presented [1, 2, 3, 4, 5, 6, 7]. These approaches can be generally classified into reconstruction-based approaches, which reconstruct high resolution image from a series of images, and learning-based ones, which usually generate high resolution image from low resolution image by learning from training sample or strong image priors learned before.

Nowadays, with rapid advance in 3D acquisition technique, 3D data are becoming more and more popular. It is of great needs of this kind of 3D super-resolution in practice, because sometimes we can't get enough information of a 3D model for rendering or recognition due to limitation of 3D acquisition system or environment condition, for example, acquisition at a long distance, capturing 3D data from non-collaborated target object.

However, most work focuses on 2D image. There is few work on 3D super resolution for 3D model as input. Our early work[11] proposed a learning-based 3D super resolution algorithm of which result is fairish but has limits such as it can not be expanded to more complex models, because what it deal with are mainly 2D data mapped from the 3D models and the mapping function can not be found in some situations.

* The authors are grateful for the grants from the National Science Foundation of China (60503019, 60533040) and Program for New Century Excellent Talents in University (NCET-04-0545).

** Corresponding author. Tel:+86-571-87951647.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 923–932, 2006.

In this paper, we propose another method for super resolution in 3D domain. When the input data is a low resolution 3D face model, the algorithm will give a high resolution version which has more detail information.

2 Algorithm

Given a low resolution face model, the purpose is to reconstruct the high resolution model which has more detail information. The new model can be represented as the combination of training samples based on the input model.

Our algorithm mainly focus on hallucinating 3D face. The whole process can be divided into two sections. The task of first section is to build the consistent mesh for all 3D face. Based on the result of first section, the hallucinating algorithm can be applied in the second section.

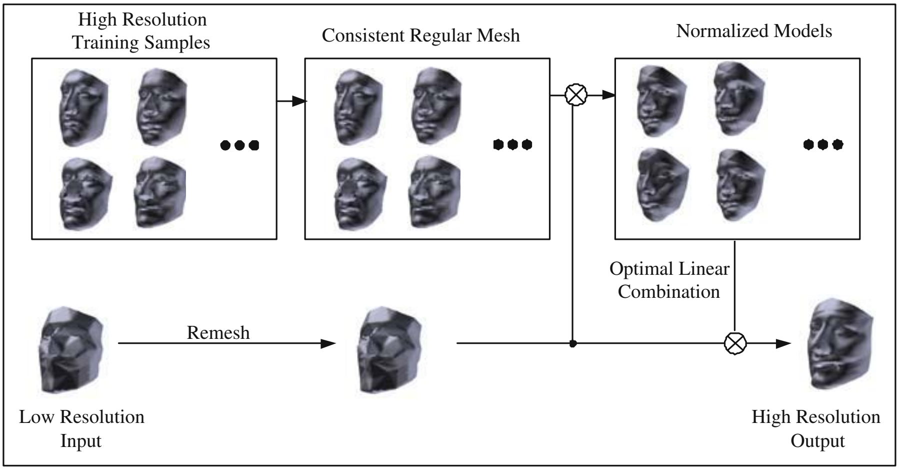
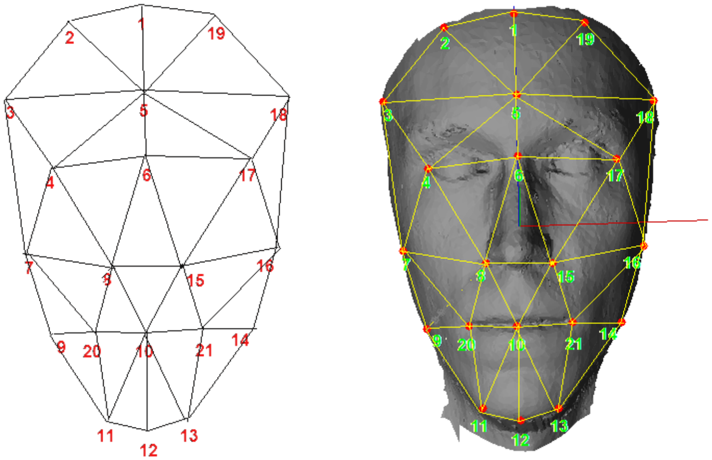


Fig. 1. The algorithm diagram

Consistent mesh. First we define a base mesh which consists of about 21 points shown in Fig.2 and manually mark the corresponding points in model we want to process. Then we use fast marching method to calculate the shortest path on the model and mark these points on the path to get the patches corresponded to the faces of base mesh. At last, we map these points in patches to the corresponding faces using area preserving mapping and subdivide the pathes to construct the consistent mesh.

Hallucinating. From the result of consistent mesh, we have built one-to-one relationship among all 3D faces including training data and input model. The hallucinating face problem could be converted to an optimal linear combination problem. To cope with limitation of shape variation in training samples, we divide the model to several patches, then the optimization of linear combination is separately solved for each patch and sew the results together to get the last result.



(a)Base mesh (b)Base mesh on 3D face

Fig. 2. The definition of the base mesh

3 Consistent Mesh for 3D Face

Given a face, we can represent it with it's original triangular mesh. But if there are more than one face model, there will be more than one mesh which does not exist the correspondence. Finding the relation between them is very important. Therefore, we need a new mesh representation called consistent mesh[8] to represent these models. It is called *consistent* because it is easy for us to find the approximate points' relationship between different faces. We define the facial base mesh and then build the consistent mesh for each given face.

3.1 Terminology

We adopt the same terminology as [8]. A triangle mesh M is a pair (P, K), where P is a set of N point P=$\{p_i = (x_i, y_i, z_i) \in R^3 | 1 \leq i \leq N\}$, and K is an abstract simplicial complex which contains all the topological information. The complex K is a set of subsets of $\{1, ..., N\}$. These subsets come in three types: vertices $\{i\}$, edges $\{i, j\}$, and faces $\{i, j, k\}$. Two vertices {i} and {j} are neighbors if $\{i, j\} \in K$. The 1-ring neighbors of a vertex $\{i\}$ form a set $V(i) = \{\{j\} | \{i, j\} \in K\}$.

3.2 Facial Base Mesh

Let common base mesh $B = (P, L)$, each vertex $p \in P$ has a corresponding point p' in the M_i as illustrated in Fig.2. The most important part of the base mesh is the topological relationship of points instead of the positions of the them. For example, we can consider the $p_8 \in P$ the right nose corner and $p_{21} \in P$ the left mouth corner, but the positions of them are not determinate. Only after all corresponding points in M_i being found, we can construct a real mesh K_0 which will be described later based on B.

We should manually label these corresponding points which are also called landmarks on the M_i. Therefore, each $p \in P$ has a determinate position and we can use these positions and the relationship extracted from B between these points to construct the mesh K_0.

If there are more than one model, we can construct a ont-to-one relationship between the points labelled in these models. But this is not enough. What we want to do is to set up correspondence of all elements among them.

3.3 Building Consistent Mesh

As describe above, we have already constructed the real mesh K_0 which has the same amount of points of base mesh B. In fact, K_0 is the consistent mesh which has then insufficiency of information compared with M_i. Obviously, this is not enough because of the insufficiency. So we must construct the consistent mesh K_N which has the enough details.

The 3D face model M_i will split into several patches, then the model will be separately remeshed. In the following paragraphs, the main operations will be explained. *Boundary tracing* finds out the boundary of the patches; *Parametrization* describes how to map the patch into a plan. In the last step *remesh*, the consistent mesh will be constructed.

Boundary Tracing. The landmarks we marked on the M_l can be denoted as p_f that means $f-$th landmarks in M_l. These landmarks in the base mesh form a group of triangle faces that can be mapped to the M_l as a group of patches of which boundaries consist of points and segments existing or being added later in the model. We called this process boundary tracing.

Given two points p_a and p_b in the base mesh and corresponding points $p_m a$ and $p_m b$ in one model. Our task is to find the points and segments or to add the points or segments that do not exist in the model. In [8], Praun proposed a method to trace a fair boundary. It is suitable for some complex conditions. But in our experiment, the points we given are limited in the face and the models do not include the afterbrain because of uselessness of their information to us. Therefore we can firstly use fast marching method [13] to get the geometry distance from $p_m a$ to $p_m b$, and then retrace from $p_m b$ to calculate the intersections of the boundary and the triangles on the model. These newly added points and edges form a boundary of a patch which we will use to form new model.

Applying this operation on each pair of points which lay on the same edge in the triangle face of base mesh will get all boundaries of patches.

Parametrization. The patches actually split the model into many small parts. Each patch can be dig up as a single object to handle. A patch P_t which consists with a set of triangles T_m have a corresponding triangle in the base mesh, so we can map all points in the P_t to the triangle. As mentioned above, the coordinates of points in base mesh do not make any sense, so we can let the coordinates of points in one triangle be any values in $x-y$ plan if them don't lay on the same line. The points on the boundary of patch P_t can simply mapped to the edge through relative position. This process is a boundary fixed mapping problem. We adopt the area preserving mapping method mentioned in[14].

We denote the triangles in the $x - y$ as T_b. Obviously there is a bijective mapping δ_p from elements in T_b to T_m. For every vertex in T_b we can calculate its real coordinate in 3D space through δ_p. And for the point p lying on a triangle $t \in T_b$, we can represent this point as the linear combination of three vertexes p_1, p_2 and p_3 of t.

$$p_i = a \times p_1 + b \times p_2 + c \times p_3 \quad (a + b + c = 1 \ and \ a, \ b, \ c \geq 0)$$

So the coordinate of corresponding point p' in P_t can be calculated with the same equation just replace the p_1, p_2 and p_3 with the corresponding points $\delta_p(p_1)$, $\delta_p(p_2)$ and $\delta_p(p_3)$. We denote this parametrization as $\rho : B \rightarrow M_l$.

Remesh. We use a method modified from [10] to remesh a model. Based on the resolution of training samples, we must adjust the J which used to control the precision of remeshed model. K^J means that perform J recursive 4-to-1 splits showed in Fig.3 of each triangle face of K_0 to represent the 0 recursive splitting. And then we find out the newly added points, which is also called knots in [10] in the M_l through function ρ. In [10], the process of remesh is divided into two types: *parametrically uniform resampling* and *geometrically uniform resampling.* We let our method restricts in the first condition because the geometrically morphing between K^0 and M_l is not so big.

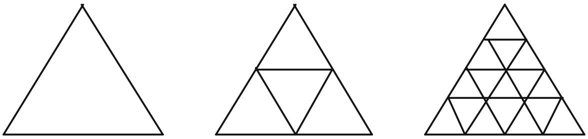


Fig. 3. 4-to-1 split in the $\rho'(M)$

4 Hallucinating 3D Face Using Consistent Mesh

4.1 Hallucinating as Optimal Linear Combination

Suppose a set of training sample M_t which has $\|M_t\|$ elements. For each model in M_t, we build the consistent mesh of it. Therefore we can represent a new model with the combination of these remeshed models which will be normalized to the same standard firstly. To a specified input low resolution model M_l, we can find out a most suitable combination for it. The process of hallucinating 3D face can be formulated as an optimization problem as following:

$$minimize \ \|M'_h - M_l\|, \qquad M'_h = \sum_{i=1}^{\|M_t\|} \alpha_i M_t^i$$

The essential of this optimization problem is searching for the shortest path problem which can be solved easily.

4.2 Divding and Sewing

If we treat $M \in M_t$ as an element in the combination, the sample number is too small to cover variation of face shape. Thus, we divide a face model M into several parts and calculate the combination of each part, then, sew the results.

Divding Face. Given the remeshed models Mt, each model $m \in M_t$ has N_{pc} patches that can be assigned to a separate coefficient which is an element in matrix C. $C_{m,n}$ is the coefficient of the n-th patches in $M_t(m)$ such that:

$$\forall\ 1 \leq j \leq N_{pc}, 1 \leq j \leq \|Mt\| \Longrightarrow C_{i,j} \geq 0$$

$$\forall\ 1 \leq j \leq P_c \Longrightarrow \sum_{i=1}^{\|Mt\|} C_{i,j} = 1$$

For an input face M_l of low resolution, we can also find the patches P_l correspond to a triangle face in base mesh and then find the corresponding patches $P_t(m)$ in $m \in M_t$, which means the bijective mapping δ from p to $p' \in P_t(m)$ could be obtained for each $p \in P_l$. Given a pair of patches p and p' such that $p' = \delta(p)$, the coordinates of them in *Cartesian Coordinate System*(CCS) have the different metric. For example, if M_l is acquired for one system which differs from the system used for training sample, the metric may be different. So we should normalize them to the same frame, using the P_l as the reference model.

The first step is scaling. For p and p' we can get the reference triangles t_r and t'_r of which vertexes can be located through base mesh. The scaling s can be approximately get from the area ratio. Then we can represent every point pt in p' as the (d, A) which d means distance from pt to t'_r and A is barycentric coordinate coefficients of pt' which is perpendicularly projecting point of p' in the plan in which t_r is located. Finally, we move the vertexes of t'_r to the corresponding vertexes of t_r and denote pt as (sd, A) which can be transformed to CCS.

Thus based on the input model M_l, a linear combination of human face is constructed. We can represent any face m_f with the linear combination through the patches P_f contained in m_f. For j-th P_f we have:

$$P_f(j) = \sum_{i=1}^{\|M_t\|} C_{i,j} P_t(i,j)$$

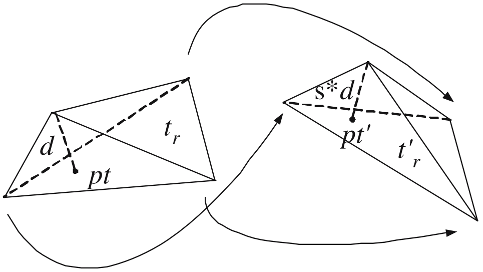


Fig. 4. Normalize patches between input data and training samples

Prior to calculate C, we define the distance function $Dist(p_1, p_2)$ as:

$$Dist(p_1, p_2) = \parallel p_1 - p_2 \parallel$$

As mentioned above, we have already normalized the training samples. So the operator $\parallel \bullet \parallel$ is a L^2 norm of differences of coordinates between p_1 and p_2. The our task is to find the coefficient C to construct face M_d in high resolution which is similar to the input model M_l, solved as following:

$$C = \arg\min_{C} \sum_{j=1}^{N_{pc}} Dist(P_d(j), P_l(j))$$

Sewing Face. Because the face we get is generated from the combination of patches of other faces. There are some slots between these patches. We blend the patches at the borders according to an algorithm proposed for images by [16]. Figure 5 shows some sewing result.

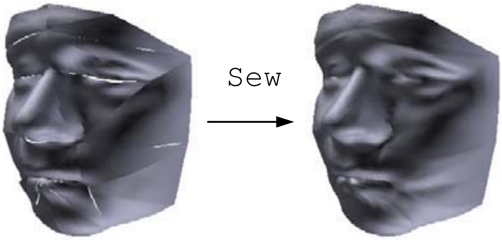


Fig. 5. Sewing Illustration

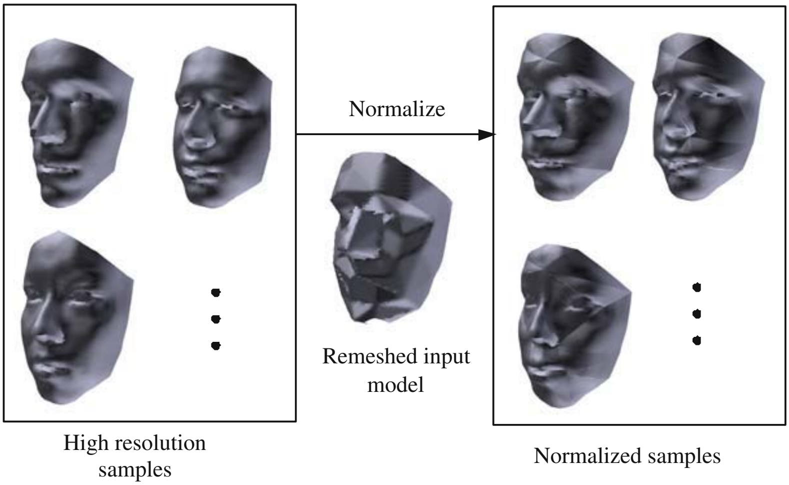


Fig. 6. The process of normalization. The left models are high resolution samples. Based on the input data, they are normalized to the right ones.

5 Simulation

Our experimental data set consists of 100 3D face models from USF Human ID 3D face database [12]. There are more than 90,000 vertices and 180,000 faces after reducing of the invalid vertices for our experiment. Considering time consuming because of the amount of data, for convenience, we use [9] to reduce the vertices of the model. We get the high resolution training models each of which has about 8,000 vertices with error tolerance 10^{-7}. The low resolution input models each of which has about 130 vertices are obtained with error tolerance 10^{-3}.

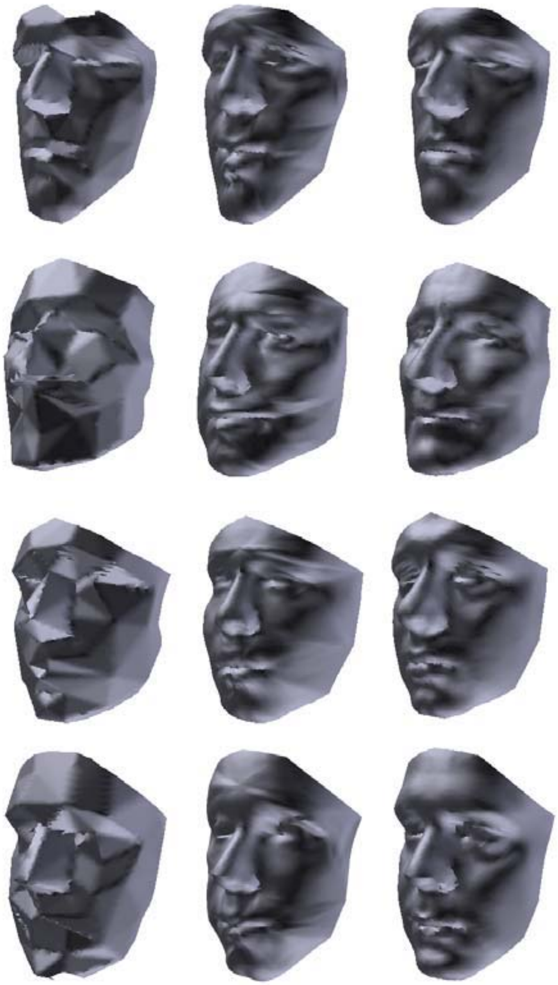


Fig. 7. Hallucinating results. The first column is the low resolution input. The middle one is reconstructed by the proposed method. The third column is the original model of high resolution.

Figure 6 depicts the normalizing step. The left models are training samples. After being normalized based on the input model, the models on the right side look like the left ones and also the input model. This operation make our latter work fall within the same frame.

Figure 7 shows the hallucinating result. The first column is the input data in low resolution, the middle one is the hallucinating result by the proposed algorithm, the third column is the original high resolution model. From the figure, our algorithm adds many details to the input data and the information added makes the input model approach the original high resolution model.

6 Conclusions

This paper has presented a super-resolution method for 3D face model, which employs the consistent mesh as the immediate form. The 3D model of hallucinating result is smoother than [11]. When no high resolution model is available in some conditions, it could give a reconstructed model with more details. It is potential to be used in many application such as rendering, editing and even recognition.

References

1. M.Elad, Y.Hel-Or,"A fast super-resolution reconstruction algorithm for pure translational motion and common space invariant blur," *IEEE Trans. on Image Processing*, 10(8):1187-1193,2001
2. N.X.Nguyen, "Numerical algorithms for image superresolution," *PhD thesis*, Stanford University, 2000.
3. F.M.Candocia, J.C.Principe, "Super-Resolution of Images Based on Local Correlations," *IEEE-NN*, 10(2):372, 1999.
4. D.Capel, A.Zisserman, "Super-Resolution from Multiple Views using Learnt Image Models," *IEEE CVPR*,pp.627-634,2001.
5. M.V.Joshi, S.Chaudhuri, "Zoom Based Super resolution Through SAR Model Fitting," *IEEE ICIP*, 2004.
6. H.Chang, D.Y.Yeung, Y.Xiong, M.V.Joshi, S.Chaudhuri, " Super-Resolution Through Neighbor Embedding," *IEEE CVPR*, 2004.
7. W.Zhao, "Super-resolution with significant illumination change," *IEEE ICIP*, 2004.
8. Praun,E., Sweldens, W. and Schroder, Peter. "Consistent mesh parameterizations," *Proceeding os SIGGRAPH'2001* , pp.179-184, 2001.
9. H.Hoppe, T.DeRose, T.Duchamp, J.McDonald, W.Stuetzle, Mesh optimization, *SIGGRAPH'93 Proceedings*, 27:19-26, 1993.
10. M.Eck, T.DeRose, T.Duchamp, H.Hoppe, M.Lounsbery, W.Stuetzle, "Multiresolution analysis of arbitrary meshes," *SIGGRAPH'95*, pp.173-182, 1995. 1995
11. S.Peng, Gang Pan, Z.Wu, "Learning-based Super-Resolution of 3D Face Model" *IEEE ICIP'05*, vol.2, pp.382-385, 2005.
12. Blanz, V., Vetter, T. " Morphable Model for the Synthesis of 3D Faces," *SIGGRAPH'99 Conference Proceedings*, pp. 187-194, 1999.
13. R.Kimmel, J.Sethian, "Fast Marching Method on Triangulated Domains," *PNAS*, 95:8341-8435, 1998.

14. M.S. Floater, "Parameterization and Smooth Approximation of Surface Triangulations," *Computer Aided Geometric Design 14* , pp.231-250, 1997.
15. A.Lee, W.Sweldens, P.Schroder, L.Cowsar, D.Dobkin, "MAPS: Multiresolution Adaptive Parameterization of Surfaces," *SIGGRAPH'98*, pp.95-104, 1998.
16. P.J. Burt, E.H. Adelson, "Mergin images through pattern decomposition," *In Applications of Digital Image Processing VIII*, no.575, pp. 173-181, 1985.
17. A.Lee, D.Dobkin, W.Sweldens and P.Schroder, "Multiresolution Mesh Morphing." *SIGGRAPH'99* , pp. 343-350, 1999.

High Quality Compression of Educational Videos Using Content-Adaptive Framework

Ankush Mittal[1], Ankur Jain[1], Sourabh Jain[1], and Sumit Gupta[2]

[1] Department of Computer Science and Enginnering,
Indian Institute of Technology, Roorkee, UA, 247667, India
{ankumfec, jainnuec, soravuec}@iitr.ernet.in
[2] Department of Computer Science and Engg.,
MVGR College of Engg., Vizianagaram, A.P., 535005, India
sumitfec@iitr.ernet.in

Abstract. With multimedia E-learning becoming popular, better compression algorithms are required that are space efficient but maintain the quality of video data, particularly slide and blackboard text. In this paper, an educational video compression technique is presented that dynamically allocates the space according to the content importance of each video segment in the educational videos. We present a phase correlation based motion estimation and compensation algorithm to encode important moving objects in efficient manner. Temporal coherence is exploited in a two phase manner. First, the frames with high similarity are categorized and encoded efficiently. Secondly, the compression ratio is adapted according to the frame content. The algorithm is compared with the state-of-the-art standards such as H.261, MPEG-4, etc. on large database. The comparison shows that for similar bit rates, the video quality for our algorithm is significantly better than the other methods.

1 Introduction

In recent years, the acquisition and indexing of rich media content has been largely automated [11], however, research challenges still remain for mass distribution of multimedia content from annotated University repositories to the learners. The challenges in instructional video streaming are the dynamic change of bandwidth (as in wireless networks), the package loss, and the differences of video content and users' preferences [2]. Some of the streaming issues have been well dealt in previous work such as [7]. This paper focuses on improving the compression of lecture videos as lack of bandwidth is one of the major bottlenecks of multimedia based distance learning. A significant research effort has been put in the last decade to achieve sufficiently high compression while retaining satisfactory video quality. MPEG-1 and MPEG-2 were introduced in the early 1990s which were based on the block based motion encoding techniques [10]. MPEG- 4, which was introduced in 1997, considers a frame to be made up of background and video objects [12]. Despite the enormous potential advantages of object based encoding, the adoption of MPEG-4 has so far been limited to the core profile - the special case of block based encoding that is very similar to the MPEG-2 standard [9].

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 933 – 942, 2006.

In this paper, we propose an efficient phase correlation based video compression technique that uses object based encoding of educational videos and provides quality videos at low bit rates. Adaptive compression is accomplished by automatically identifying the importance of content of each frame and changing the bit rate according to the content priority. Also, temporal redundancy is avoided in consecutive frames that have very little change. The experimental results and comparison with the state-of-the-art compression algorithms such as MPEG-1, MPEG-2, RealMedia, etc., show the effectiveness of our approach. A typical one hour video data can be compressed to as low as 20 Megabytes using our algorithm.

2 Related Works

Using video for educational purposes is a topic that has been addressed at least since 1970s [8]. Recently, several universities have started offering entire degree programs based on transmission of lecture videos. A case in point is Singapore-MIT Alliance (SMA) development program[1]. SMA lectures are given daily, and it is expensive to process, index and label them through manual methods. The focus of the research in educational video data has been significantly on content creation and indexing. Ip and Chan in their Automatic Segmentation and Index construction for the Lecture Video [11] use the lecture note along with Optical Character Recognition (OCR) techniques to synchronize the video with the text. Content-based adaptive streaming system has been designed for baseball and other sports videos [1]. In a similar work for low bit-rate video streaming for face-to-face teleconferencing, the input video frame is first processed to allocate the face and its components. Although these works have shown significant improvement in compression performance, they cannot be applied directly to E-learning video.

3 System Overview

Figure 1 presents a high level overview of our entire video compression system. In the first step, a content based classification of the frames is performed. Each frame is classified into one of the four classes, which can be transmitted at different bit rates. Next, the blackboard is extracted from these frames. Then, a model of the background without the teacher is created using input from multiple segmented frames. This in turn aids in segmenting the teacher in further frames. It can be safely assumed that for this part of video processing, the changes in the background are unimportant. The changes in the blackboard are captured through an efficient binary encoding technique, and an efficient phase correlation based motion estimation algorithm is used for transmitting the teacher data over the channel. Judicious handling of redundant data aids in increasing the compression performance while providing high quality videooutput. At the receiver side, all the frames are reconstructed in real time using the transmitted data.

[1] Web.mit.edu/sma/

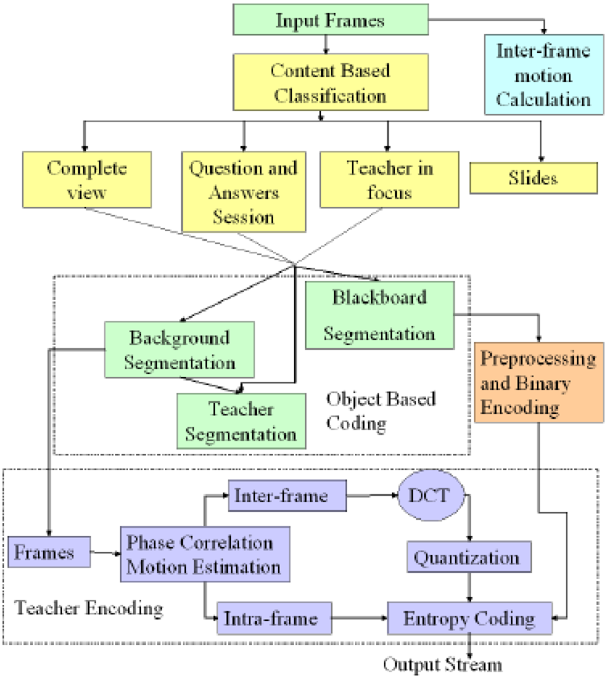


Fig. 1. Overview of the educational video compression system

4 Content Based Classification

Content based video classification can be used to divide the video into various segments depending upon their contents and transmit these segments at varying bit-rates. Scenes containing slides and question answer session can be transmitted at lower bit rates in an efficient manner, whereas close up scenes with teacher in focus and his expressions being very prominent may require higher bit rates than normal full view scenes. Typically, in a classroom video, we can have four categories of shots - a full view of the classroom, a close-up of the teacher, slides and a typical shot in a question answer session [3]. Here, we discuss how all these scenes can be identified and segmented out from an educational video.

4.1 Detecting a Full Class View

Full view classroom frames have some typical characteristics such as the complete board is shown, some students may be visible in the view and a significant part of the teacher's body may be visible in the frame. Thus, a frame can be classified in the above category if it satisfies the following two conditions: The board should be present and completely detected and the computed area representing the teacher should be less than one-third of the entire frame.

4.2 Detecting Slides

Slides can be detected by computing the motion between consecutive frames. If the motion is almost zero for a couple of seconds, then the frames are labelled as slides. The frames corresponding to the slides do not change over time, so all we need to transmit in case of slides is the initial frame and the time for which the slide persists.

4.3 Detecting Close Up Scene and Question Answer Session

If a scene has not been classified in the above two classes, either it should be the close up scene with teacher in focus or a Question Answer session. In either case, the complete board would not be visible. The scene is classified as a close up scene if the area containing teacher is quite large, typically around half of the overall frame size and the number of edges detected in such frame are very small. If the teacher's area is small and the number of edges is large, the scene is classified as a Question Answer scene. Our algorithm for detection of this category is simpler than the algorithm proposed by Li and Dorai [6].

5 Object Segmentation and Compression

In this module, we segment all frames into their constituent components, that is, the teacher, the blackboard and the background.

5.1 Board Segmentation

For detecting the blackboard in any frame (Figure 2a), the Canny edge detector is applied for finding the horizontal and vertical edges. A typical set of horizontal and vertical edges detected by applying a Canny edge detection algorithm on a frame containing the complete view of the class are shown in Figures 2b and 3a, respectively. Next, we perform the dilation of the horizontal edges using a vertical element of 5-6 pixel length, and the dilation of the vertical edges using a horizontal element of the same dimensions. We obtain two different images containing the horizontal edges and the vertical edges as shown in Figures 2c and 3b. It may be observed that the horizontal and vertical edges may not necessarily correspond to the board but may be due to the walls. The correct region representing the board has to be selected.

Next, the largest horizontal edge detected in Figure 2c is chosen and the longest vertical edge close to its end points is detected in Figure 3b. Then a rectangular region is constructed using these two edges that may represent a blackboard region. Several

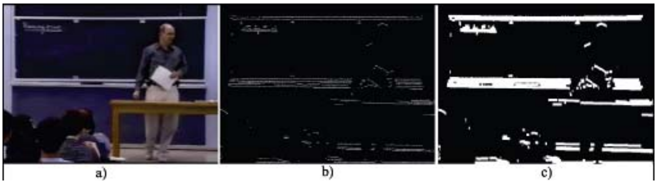


Fig. 2. a) The original frame b) The horizontal edges detected c) The dilated horizontal edges

rectangular regions are chosen within the blackboard and the average intensities are computed in these regions to verify that the intensity values correspond to the blackboard region. In this method, if the verification provides negative results, the horizontal edge is discarded and the next largest horizontal edge is chosen. Figure 3c shows the board detected (the patterned region) using images Figure 2c and 3b.

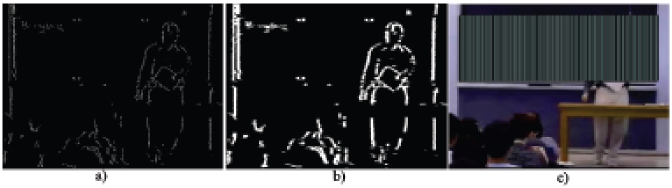


Fig. 3. a) The vertical edges that have been detected, b) The dilated vertical edges and c) The patterned blackboard that has been detected

5.2 Teacher Segmentation

We propose a pixel-ratio comparison based robust algorithm that accurately segments the teacher. For teacher segmentation first the background is modelled, and then the teacher can be tracked in any of the subsequent frames.

5.2.1 Background Modelling

In most videos, the reference image of a classroom without the teacher is not available. To model the background, we need to subtract the region corresponding to the teacher [5]. For the initial prediction of the background, we assume a set of frames that have significant teacher motion. This can be accomplished by taking frames that are widely separated in time. Let the two frames to be compared be represented as frames f(i) and f(j). For each pixel (x,y) in f(i) and f(j), we compute the ratio of the intensity values. We cluster all the pixels where the ratio is smaller than a prespecified threshold. Small and noisy clusters are eliminated through morphological operations viz. erosion and dilation. Next, we enclose the clusters with rectangular boxes and discard the boxes that do not have sufficient pixels to represent a teacher. This results in two possibilities: either the rectangular regions overlap in two frames or they have no common regions. We discard the cases of images where the rectangles overlap because they cannot be used for reconstructing the background information completely. The entire background can be modelled when we come across a case with two large rectangles which are not overlapping. In this case the entire background can be reconstructed by simple masking of the teacher and combining the regions of the two frames.

5.2.2 Predicting the Region of the Teacher

It can be done by simple comparison of the given frame with the background frame. For each pixel in the two frames we compute the ratio of the intensities. If the intensity ratio is larger than one, we interchange the numerator and denominator. If the resultant ratio is smaller than a specified threshold (typically 0.6-0.8), it is inferred that a significant change has taken place at that pixel and it belongs to a region of moving object - the teacher. The results over several frames show that this technique is a robust one for teacher segmentation.

5.3 Board and Background Transmission

After the segmentation of each frame into its three components, we can deal separately with these objects. The background needs to be transmitted infrequently as it is usually static. The changes in the blackboard have to be transmitted from time to time. The region corresponding to the teacher is masked and the remaining portion of the blackboard is visible in each view. A binary region transmission of the blackboard is performed. First, a binary thresholding of the blackboard region is done. Next, regions which have values smaller than the threshold are taken to be black regions and those above the threshold are assumed to be white. The contents of the blackboard region get enhanced by the above process because of better contrast.

6 Phase Correlation Based Motion Estimation

There are three main types of motion estimation methods: block matching methods, phase correlation methods, and gradient based methods. The phase correlation motion estimation has much lower complexity as compared from other methods it measures the motion directly from phase correlation, and gives much smoother motion vector field. The major overhead for our algorithm is the transmission of the teacher's image. We use the phase correlation based motion estimation for compression of the image sequences containing the teacher. Due to its advantage of immunity to overall illumination changes and noise, it provides a better video compression.

In this technique, we compute the displacement of a pixel directly by using the phase information.

The objective is to compute the displacements Δx and Δy of all pixels (x,y) in a frame f(x,y) at time t1 to time t2, with the condition that:

$$f(x, y, t1) = f(x + \Delta x, y + \Delta y, t2) \quad (1)$$

Taking Fourier transform F(u, v) of an image f(x, y) between these two frames, we get:

$$F(u, v, t1) = F(u, v, t2)e^{2j\prod(\_\Delta x\times u+\Delta y\times v)} \quad (2)$$

From Equation 2 it is clear that any translation motion between two frames is depicted as a phase change in the frequency domain. In order to determine the interdependence of two frames, we compute the cross-correlation between them. The cross correlation function, $R_{(x,y)}(t1, t2)$ provides a means of quantifying the interdependence of the two signals. The cross correlation function can be represented as:

$$R_{(x,y)}(t1, t2) = E[f(t1)f(t2)] \quad (3)$$

Where E[X(t)] is the expected value of X(t). Hence, in the frequency domain, its equivalent, the power spectral density $S(\omega)$, is given by the following convolution operation:

$$S(x,y)(\omega) = F(u, v, t1) * F^{*}(u, v, t2) \quad (4)$$

Normalized power spectral density, $SN_{(x,y)}(\omega)$, is used to remove any luminance variation. Using Eq. 2 and 4 , we have:

$$SN(x,y)(\omega) = \exp[-2j \prod (\Delta x\, u + \Delta y\, v)] \tag{5}$$

Note that the phase value of $SN_{(x,y)}(\omega)$ is equal to the conjugate of the phase value obtained in Eq. 2. This relationship can be used to compute the displacement by taking the inverse Fourier transform of normalized power spectral density which gives us the $R_{(x,y)}(t1, t2)$ as:

$$R_{(x,y)}(t1, t2) = \delta(x - \Delta x, y - \Delta y) \tag{6}$$

This δ function corresponds to the displacement of a pixel from one frame to the next. To find a better correlation between adjacent frames, we dilate a block of size say nxn to block of size (2n) × (2n). A weighted raised cosine transform is applied to this (2n)×(2n) size block to assign more weightage to our original n × n size block [13]. In order to get better results we use Half-Pixel Motion Estimation method to improve the estimation of motion vectors. After computing all the displacement vectors, we reconstruct the expected frame. Next, the peak signal to noise ratio (PSNR) is computed by comparing the reconstructed frame with the original frame. Similar to the MPEG B-frames, P-frames and I-frames, we use the A-frames, B-frames and the C-frames. The A-frames are reconstructed using only the motion vectors, the teacher data in the B-frames is compressed using the phase correlation based motion estimates and residual errors. In the C-frames the teacher region is compressed using only the JPEG compression techniques. The C-frames are inserted in the regions where the PSNR values of the reconstructed frames fall below λ_1. The B-frames are required when the PSNR values fall between λ_1 and λ_2, and for PSNR values greater than λ_2, we transmit the A-frames.

7 Experimental Results

We conducted experiments on several videos available as a part of the Singapore-MIT Alliance program, and those recorded at our Institute for testing the compression performance. In this section, the results for three video sequences are presented. The original videos are of 90 minutes duration corresponding to one lecture duration. For the purpose of brevity and illustration of working of our algorithm, we present the results computed over 4800 frames of video. The characteristics of these three video sequences are given in Table 1. Type 1 scenes are the complete view of the class, Type 2 scenes represent the close up views where the teacher is in focus, Type 3 scenes are the slides and the Type 4 scenes are the Question and Answer sessions.

Table 1. Results of Content Based Classification Algorithm for different video samples

Sequence	Source	Type1	Type2	Type3	Type4
Video 1	MIT	3050	400	950	400
Video 2	IITR	3200	650	700	250
Video 3	MIT	2750	975	1075	0

Accuracy of the content-based classification algorithm for the Type 4 (slides) is 100% whereas accuracy varies in the range of 96-100% for other frames.

We compare our algorithm(OA) with MPEG-1, MPEG-2, MPEG-4 and H.264. In order to do the performance comparison, videos of 20 MB size were generated by modifying the parameters of MPEG-1, MPEG-2, and other algorithms. The PSNR values for different frame types(FT) were computed for different algorithms and are tabulated in Table 2. Since we store the original slide (type 3 frame) in high resolution, the PSNR value of type 3 frame is infinite for our algorithm. It can be observed that for all types of frames, our algorithm gives significantly higher PSNR performance.

Table 2. Comparison of PSNR value obtained from different algorithms for 20MB size videos. Higher PSNR value indicate better quality.

Frame Type	Our Algorithm	Mpeg1	Mpeg2	Mpeg4	H.264
1	31.8	23.1	22.3	23.4	23.6
1	31.9	22.9	22.9	22.8	23.0
1	32.2	22.9	23.2	23.9	23.7
2	33.5	27.9	27.9	28.0	28.1
2	34.8	27.9	27.9	27.9	27.9
3	Infinite	21.0	20.9	21.1	21
4	31.4	20.9	20.9	20.7	20.9

For lecture video compression, it is especially required that the text quality of the blackboard and the slides in the video is good. In order to evaluate the text legibility, we used luminance measure [4]. Luminance measure L is defined as:

$$L = 0.3R + 0.59G + 0.11B. \tag{7}$$

where *R, G*, and *B* are red, green and blue respectively. A strong, sharp contrast of luminance levels between text and background makes text readable. Compressing text tends to smear out this sharp contrast, bringing the luminance of background and text closer to the middle luminance levels, while creating a spread of luminance levels in-between. Plotting the frequency of each luminance level in a region of text on a curve, one would expect to see two modes, two strong local maxima, representing the text and the background. And when the text is blurred by compression, the strength of these modes is decreased, and they move closer together, reducing the contrast of the

Table 3. Comparison of luminance values obtained for different algorithms over 20 MB video. Higher luminance value indicates better quality. The luminance is used to evaluate the slide quality.

Original	Our Algorithm	Mpeg1	Mpeg2	Mpeg4	H.264
6.8353	6.8353	2.0157	1.6941	2.5098	1.70
5.7137	5.7137	1.6471	1.4392	2.2176	1.82
6.7922	6.7922	2.0824	2.0549	2.1490	2.02
8.9176	8.9176	1.8745	1.5373	2.5569	1.71

text. Table 3 presents the comparison of luminance values for different algorithms and a comparison of the quality of the frames for different techniques of compression is shown in the fig.4. Our algorithm preserves the slide quality for the entire video and thus has the same luminance value as the original uncompressed video. The luminance value of MPEG-4 is second best, though significantly less than that of our algorithm.

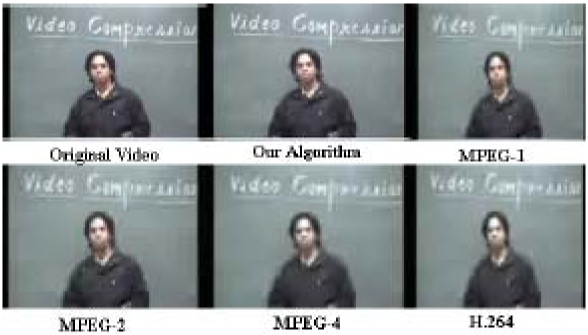


Fig. 4. Comparison of different compression algorithms for video size of 20 MB

Table 4. Comparison of performance of different algorithms in preserving edges of the blackboard text. The figures correspond to number of edges detected using Canny edge detector.

Original	Our Algorithm	Mpeg1	Mpeg2	Mpeg4	H.264
78	78	25	25	21	20
80	80	31	27	28	25
78	78	29	29	29	28
75	75	28	38	47	42
%Edge Preservation	**100%**	**39.5%**	**38.3%**	**40.2%**	**37.0%**

Another measure for evaluating the readability of lecture videos is preservation of edges. Table 4 presents a comparison of different algorithms over the frame which consists of blackboard with text written on it. In our algorithm, we threshold the blackboard text and thus all the edges are preserved, while most of the edges are lost in other algorithms.

8 Conclusions and Future Work

In this paper, a strategy for content-adaptive video compression has been proposed. The key contribution of the paper is to provide a generic algorithm that significantly reduces the lecture video size, while maintaining the resolution of important video segments. The phase correlation based motion encoding of the moving objects provides high compression ratios. Robust algorithms for teacher and blackboard segmentation results in accurate reconstruction of the scene during decoding.

A significant improvement could be accomplished by creating and transmitting the templates of teacher's hand motions and facial expressions, as was done in face-to-face teleconferencing [15]. The technique presented in this paper can be extended to slow-moving structured videos, such as News, interview videos etc., for achieving high compression.

References

[1] D. Zhong and S.-F. Chang, "Structure Analysis of Sports Video Using Domain Models," *IEEE Conference on Multimedia and Exhibition, Tokyo, Japan*, pages 182-185, 2001.

[2] T. Liu and J. R. Kender, "Lecture videos for E-learning current research and challenges," *IEEE International Workshop on Multimedia Content-based Analysis and Retrieval*, 2002.

[3] Z. Zhu and C. McKittrick and W. Li, "Virtualized Classroom Automated Production, Media Integration and User-Customized Presentation," *Workshop on Multimedia Data and Document Engineering (with CVPR)*, 2004.

[4] M. Eckert and A. Bradley, "Perceptual Models Applied to Still Image Compression," *Signal Processing*, , pages 177-200, vol. 70, 1998.

[5] Y. Ming and J. Jiang and J. Ming, "Background Modeling and Subtraction Using a Local- Linear- Dependence-Based Cauchy Statistical Model," *Proc. of Digital Image Computing: Techniques and Applications*, pages 469-478, 2003.

[6] Y. Li and C. Dorai, "Detecting discussion scenes in instructional videos," *International Conference on Multimedia and Expo*, , pages 13`1-1314, 2004.

[7] T. Liu and C. Choudary, "Real-time Content Analysis and Adaptive Transmission of Lecture Videos for Mobile Applications," *Proceedings of ACM Multimedia*, pages 400-404, 2004.

[8] D. A. Michalopoulos , "A video disc oriented educational system," *ACM SIGCSE-SIGCUE technical symposium on computer science and education*, pages 389-392 , Feb 1976.

[9] O. Avaro and P. A. Chou and A. Eleftheriadis and C. Herpel and C. Reader and J. Signes., "The MPEG-4 systems and description languages: a way ahead in audio visual representation," *Signal Processing and Image Communication, Special Issue on MPEG-4*, pages 385-431 , 1997.

[10] D. LeGall , "MPEG: a video compression standard for multimedia applications," *Communications of the ACM*, pages 46-58 , April 1991, Vol 34, No 4, 13

[11] H. H. S. Ip and S. L. Chan "Automatic Segmentation and Index Construction for lecture video," *Journal of Educational Multimedia and Hypermedia 7(1)*, pages 91-104, 2004

[12] I. Richardson "H.264 and MPEG-4 Video Compression: Video Coding for next-generation multimedia" , Wiley Publishing. 2003

[13] S. Haykin "Communication systems" , John Wiley and Sons, New York, 2001.

Double Regularized Bayesian Estimation for Blur Identification in Video Sequences

Hongwei Zheng and Olaf Hellwich

Computer Vision & Remote Sensing, Berlin University of Technology,
Franklinstrasse 28/29, Office FR 3-1, D-10587, Berlin
{hzheng, hellwich}@cs.tu-berlin.de

Abstract. Blind blur identification in video sequences becomes more important. This paper presents a new method for identifying parameters of different blur kernels and image restoration in a weighted double regularized Bayesian learning approach. A proposed prior solution space includes dominant blur point spread functions as prior candidates for Bayesian estimation. The double cost functions are adjusted in a new alternating minimization approach which successfully computes the convergence for a number of parameters. The discussion of choosing regularization parameters for both image and blur function is also presented. The algorithm is robust in that it can handle images that are formed in variational environments with different types of blur. Numerical tests show that the proposed algorithm works effectively and efficiently in practical applications.

1 Introduction

The primary goal of blind image deconvolution (BID) is to recover lost information from a degraded image for obtaining the best estimate to the original image. Its applications include photography debluring, remote sensing, medical imaging, and multimedia processing. An ideal image f in the object plane is normally degraded by a linear space-invariant point spread function (PSF) h with an additive zero mean Gaussian white noise n using $g = hf + n$. The equation provides a good working model for image formation. An observed image in the image plane g is formed by two unknown conditions h and n. The two-dimensional convolution can be expressed as $hf = Hf = Fh$, where H and F are block-Toeplitz matrices and can be approximated by block-circulant matrices for large images.

In two decades, there are has been considerable interest in the regularization theory. A regularization method is originally proposed by Tikhonov [1], Miller [2] et al. which replaces an ill-posed problem by a well-posed problem with an acceptable approximation to the solution. Later, Katsaggelos et al. [3] have introduced an iterative regularization algorithm for image restoration based on a set theoretic approach. This algorithm uses a deterministic framework to introduce *a priori* knowledge in the form of convex sets, and to decouple the nonlinear observation model into double linear observation models that are easy to solve.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 943–952, 2006.

A projection-based method with conjugate-gradient minimization for BID has been proposed and extended by [4], [5], [6]. These methods have demonstrated how the parametric models in image restoration methods are used [7], [8] in some respects. However, these results are observed in underutilization of prior information. The ill-posed image restoration problem needs more effective prior information or constraints to yield a unique solution to the corresponding optimization problem. Even if a unique solution exists, a proper initialization value is still intractable, e.g. cost function is non-convex.

The Bayesian estimation provides a structured way to include prior knowledge concerning the quantities to be estimated [9], [10]. The Bayesian approach is, in fact, the framework in which the most recent restoration methods have been introduced. When blur is present, different approaches have been proposed to find a maximum a posterior (MAP) estimate. Blake et al. [11] propose the use of gradually non-convexity method, which can be extended to the blurring problem. Molina and Ripley [12] propose the use of a log-scale for the image model. Green [13] and Bouman et al. [14] use convex potentials in order to ensure uniqueness of the solution. Recently, an appreciable extension of the range of hyperparameter estimation methods is used in Bayesian estimation. Molina et al. [15] use a hierarchical Bayesian paradigm resulting from the set theoretic regularization for estimating hyper-parameters. They also report that the accuracy of the obtained statistic estimates for the PSF and the image could vary significantly, depending on the initialization. To obtain accurate restorations in the Bayesian approach, accurate prior knowledge of PSF or image must be available.

In this paper, a space-adaptive regularization method is integrated into a Bayesian learning approach. A newly introduced solution space of PSF priors supports accurate parametric PSF in the form of Bayesian MAP estimation. An integrated quadratic cost function subject to convex constraints is minimized by projecting iterations onto an alternating minimization within a specified range. These positivity constraints and strictly convex property ensure that the alternating minimization procedure converges globally. Although the convergence solution depending on the initial value [16], the estimated PSF values support accurate initial value. Regularization parameters and weight matrices are estimated with the help of L-curve technique [17].

The paper is organized as follows. In Sect. (2), Bayesian estimation in the context of double regularized iterations is described. In Sect. (3), the proposed cost functions are optimized in alternating minimization. Experimental results are shown in Sect. (4). Conclusions are summarized in Sect. (5).

2 Bayesian Estimation in Double Regularizations

The Bayesian MAP estimation utilizes a prior information to achieve a convergent posterior. Following the Bayesian paradigm, the true $f(x)$, the PSF $h(x)$ and the observed $g(x)$ are formulated in

$$p(f, h|g) = p(g|f, h)p(f, h)/p(g) \propto p(g|f, h)p(f, h) \tag{1}$$

Applying the Bayesian paradigm to the blind deconvolution problem, we try to compute convergence values from Eq. (1) with respect to $f(x)$ and $h(x)$. This Bayesian MAP estimation can also be seen as a regularization approach which combines the optimization method to minimize two proposed cost functions in the image domain and the PSF domain. The cost function of the true image $f(x)$ and the cost function of the PSF $h(x)$ are deducted from Eq. (1), then we get $L(\hat{f}_{(g,h)}) \propto p(g|\hat{f},h)p(\hat{f})$, $L(\hat{h}_{(g,f)}) \propto p(g|f,\hat{h})p(\hat{h})$. For the application of these equations, some constraints are assumed due to the fact that the image pixels are independent identically distributed and do not influence the pixel correlations.

2.1 Weighted Space-Adaptive Regularization

The direct least squares solution is $\sum_{x\in\Omega}(h(x)*f(x)-g(x))^2 = \min$. This equation may lead to a vector $f(x)$ that is severely contaminated with noise. A Tikhonov regularization [2], [1] can efficiently solve such ill-posed inverse problem with additive noise. This equation adds a penalty term L^2 norm of the image f multiplied by a regularization parameter λ for solving the linear least squares problem, $\frac{1}{2}\sum_{x\in\Omega}(h(x)*f(x)-g(x))^2+\frac{1}{2}\lambda\sum_{x\in\Omega}f(x)^2 = \min$. However, some ringing artifacts near sharp intensity transitions are still attributable to the Tikhonov regularization. To reduce the ringing effects, Lagendijk et al. [18] made an extension of it by making use of the theory of the projections onto convex sets [3], and the concepts of norms in a weighted Hilbert space. A weighted space-adaptive regularization equation then seeks to minimize the following cost function as shown in Eq. (2),

$$\frac{1}{2}\sum_{x\in\Omega}w_1(h(x)*f(x)-g(x))^2+\frac{1}{2}\lambda\sum_{x\in\Omega}w_2(c(x)*f(x))^2 = \min \tag{2}$$

where the cost function is minimized based on the degraded image $g(x)$, the ideal image $f(x)$, and the PSF $h(x)$. $c(x)$ is a regularization operator. λ is a regularization parameter that controls the trade-off between the fidelity to the observation and smoothness of the restored image. Normally, real images are piecewise smooth and additive noise is not spatially stationary. The trade-off should be spatially adaptive according to the local properties of image and noise. The ringing artifacts and noise magnification can be roughly controlled by λ firstly. Weights w_1 and w_2 can then reduce these two effects adaptively to achieve better visual evaluation.

2.2 Solution Space of Blur Kernel Priors

We define a set Θ as a solution space of Bayesian estimation which consists of primary parametric PSF models as $\Theta = \{h_i(\theta), i = 1,2,3,...,N\}$. $h_i(\theta)$ represents the ith parametric PSF with its own parameters θ, and N is the number of PSFs.

$$h_i(\theta) = \begin{cases} h_1(\theta) \propto h(x,y;L_i,L_j) = 1/K, & \text{if } |i| \le L_i \text{ and } |j| \le L_j \\ h_2(\theta) \propto h(x,y) = K\exp(-\frac{x^2+y^2}{2\sigma^2}) & \\ h_3(\theta) \propto h(x,y,d,\phi) = 1/d, & \text{if } \sqrt{x^2+y^2} \le D/2, \tan\phi = y/x \end{cases} \tag{3}$$

$h_1(\theta)$ is a Pillbox blur kernel with a length of radius K. $h_2(\theta)$ is a Gaussian PSF and can be characterized by parameters with its variance σ^2 and a normalization constant K. $h_3(\theta)$ is a simple linear motion blur PSF with a camera direction motion d and a motion angle ϕ. The other blur structures like out-of-focus and uniform 2D blur [19], [7] are also built in the solution space as *a priori* information. The solution space is then constructed by a set of predefined parametric PSFs for estimation in the Bayesian MAP Estimation.

2.3 Estimation in the Image Domain and the PSF Domain

In the image domain, the cost function of image estimate can be minimized iteratively in the weighted space-adaptive regularized formulation. In this equation, $p(g|\hat{f},h)$ follows a Gaussian distribution and $p(f)$ is prior knowledge with some constraint conditions.

$$\begin{aligned} L(\hat{f}_{(g,h)}) &= \arg\max_{\hat{f}}[p(g|\hat{f},h)p(\hat{f})] \qquad (4)\\ &= \frac{1}{2}\sum\nolimits_{x\in\Omega} w_1(g(x)-h(x)*f(x))^2+\frac{1}{2}\lambda\sum\nolimits_{x\in\Omega} w_2(c_1(x)*f(x))^2 \end{aligned}$$

where $p(g|\hat{f},h)\propto \exp\left\{-\frac{1}{2}\sum_{x\in\Omega} w_1(g(x)-h(x)*f(x))^2\right\}$ and the prior of image is $p(\hat{f})\propto \exp\left\{-\frac{1}{2}\lambda\sum_{x\in\Omega} w_2(c_1(x)*f(x))^2\right\}$. The first term is a fidelity term and the second is a smoothing term. Direct minimization of the cost function would lead to excessive noise magnification due to the ill conditioning of blur operator. A smoothness constraint $c_1(x)$ is an regularization operator and usually is a high-pass filter.

In the PSF domain, PSF can be seen as maximizing the conditional probability. However, manipulation of probability density functions (PDF) of PSFs in Bayesian estimation is difficult. A PSF estimation of a given image must be made firstly to attribute an accurate initial value in the regularization. The proposed prior solution space supports the parametric structured PSFs in Bayesian estimation. One more cost constraint for the estimated PSF is then added in the equation. A new cost function for PSFs is following:

$$\begin{aligned} L(\hat{h}_{(g,f)}) &= \arg\max_{\hat{h}}\left\{p\left(g\left|\hat{h},f\right.\right)p_\Theta\left(\hat{h}\right)\right\}=\frac{1}{2}\sum\nolimits_{x\in\Omega} w_1(g(x)-h(x)*f(x))^2\\ &+\frac{1}{2}\beta\sum\nolimits_{x\in\Omega} w_3(c_2(x)*h(x))^2+\frac{1}{2}\gamma\sum\nolimits_{x\in\Omega} w_4|\hat{h}-\hat{h}_f|^2\} \qquad (5) \end{aligned}$$

where $p_\Theta(\hat{h})\propto \exp\left\{\frac{1}{2}\beta\sum_{x\in\Omega} w_3(c_2(x)*h(x))^2\right\}+\exp\left\{\frac{1}{2}\gamma\sum_{x\in\Omega} w_4|\hat{h}-\hat{h}_f|^2\right\}$ is the prior knowledge and need to be first computed . $\hat{h}$ is the current PSF of a given image and $\hat{h}_f$ is the final result of PSF for this given image. Since both the ideal and the observed image represent nonnegative intensity distributions, the PSF coefficients are $h(x)\geq 0$. Furthermore, the image formation system normally does not absorb or generate energy, the PSF satisfies $\sum_{x\in\Omega}\mathrm{h}\,(\mathrm{x})=1.0$. The probability of the current PSF is computed in a Gaussian distribution density,

$$h_i(\theta^*) \propto \arg\max_{\theta} \log p\left(h_i(\theta)\left|\hat{h}\right.\right) \tag{6}$$
$$= \arg\max_{\theta} \log\left\{\frac{1}{(2\pi)^{\frac{LB}{2}}\left|\sum_{dd}\right|^{\frac{1}{2}}}.\exp\left[-\frac{1}{2}\left(h_i(\theta)-\hat{h}\right)^T \sum\nolimits_{dd}^{-1}\left(h_i(\theta)-\hat{h}\right)\right]\right\}$$

We define the likelihood of the current PSF $\hat{h}$ and in resembling the ith parametric model $h_i(\theta)$, $h_i(\theta) \in \Theta$. The first subscript i denotes the index of the blur kernel. The modeling error $d = h_i(\theta) - \hat{h}$ is assumed to be a zero-mean homogeneous Gaussian distributed white noise process with covariance matrix $\sum_{dd} = \sigma_d^2 I$ independent of image $f(x, y)$. LB is the support size of the blur. The likelihood of the current PSF $l_{ij}(\hat{h})$ is computed using a Euclidean distance between the current PSF $\hat{h}$ and the corresponding probability model $h_i(\theta^*)$, $l_{ij}(\hat{h}) = \sum_{i=1}^{N} \exp\{-|h_i(\theta^*) - \hat{h}|^2/[2tr(\sum_{dd})]\}$. Using the K-NN concept [9], we use a weighted mean filter to find the likelihood of $\hat{h}$ belonging to the ith parametric blur model. The mean value of likelihood $l_m(\hat{h})$ is $l_{ij}(\hat{h})$ weight-divided by $d(\hat{h}, \hat{h}_j)$. $d(\hat{h}, \hat{h}_j)$ is the Euclidean distance between $\hat{h}$ and its neighbor $\hat{h}_j$. The weighted mean likelihood $l_m(\hat{h})$ should depend on the likelihood value of the blur manifold $l_{ij}(\hat{h})$ and the distance between $\hat{h}$ and its neighbor PSF $\hat{h}_j$. The final PSF $\hat{h}_f$ is obtained from the parametric PSF models using $\hat{h}_f = [l_0(\hat{h})\hat{h} + \hat{h}_i(\theta^*)\sum_{m=1}^{C} l_m(\hat{h})]/[\sum_{m=1}^{C} l_m(\hat{h})]$, where $l_0(\hat{h}) = 1 - max(l_m(\hat{h}))$, $m = 1, ..., C$. The optimal parametric model $\hat{h}_i(\theta^*)$ is computed based on the estimated $\hat{h}$. In reality, most blurs satisfy up to a certain degree of parametric structure. The main objective is to assess the relevance of current blur $\hat{h}$ with respect to parametric PSF models $h_i(\theta)$, and integrates these prior knowledge progressively into the computation scheme. If the current blur $\hat{h}$ is close to estimated $\hat{h}_f$, that means $\hat{h}$ belongs to a parametric blur structure. Otherwise, the current blur $\hat{h}$ may to not belong to the predefined PSF priors.

3 Alternating Minimization

The objective of the convergence procedure is to minimize double cost functions by combining the cost functions of image and PSF. These two L^2 norm regularizations are shown to be quadratic with positive semi-definite Hessian matrices. The two cost functions are convex functions which ensures the existence, uniqueness and stability of the convergence value in their respective domains. We propose to solve the equation as follows:

$$\min_{\hat{h},\hat{f}} L(\hat{f},\hat{h}) = \frac{1}{2}\sum\nolimits_{x\in\Omega} w_1(g(x) - h(x) * f(x))^2 + \frac{1}{2}\lambda\sum\nolimits_{x\in\Omega} w_2(c_1(x) * f(x))^2$$
$$+\frac{1}{2}\beta\sum\nolimits_{x\in\Omega} w_3(c_2(x) * h(x))^2 + \frac{1}{2}\gamma\sum\nolimits_{x\in\Omega} w_4(\hat{h} - \hat{h}_f)^2 \tag{7}$$

The resulting method attempts to minimize double cost functions subject to constraints such as non-negativity conditions of the image and energy preservation of PSFs. During the implementation, λ, β, γ including diagonal matrices assign different emphases on the balance of the convergent PSF and image. The weights are calculated according to [3], [6], [18]: $w_1 = 1$, if data at x is reliable, otherwise $w_1 = 0$; the image weight $w_2 = 1/[1+\alpha_2\hat{\sigma}_f^2(x)]$, $\hat{\sigma}_f^2(x)$ is local variance of the observed image at x in a given window, and $\alpha_2 = 1000/\sigma_{max}^2$ is a tuning parameter designed so that $w_2 \rightarrow 1$ in the uniform regions and $w_2 \rightarrow 0$ near the edges. As to the weight of PSF, we note that the initial PSF is estimated previously, we take $w_3 = 1$, $w_4 = 1$. The cost function of this equation is minimized in an alternating optimization approach via conjugate gradient descent.

The alternating minimization (AM) decreases complexity. Derived from Eq. (7), we get two partial differential equations $p(x) = \partial L(\hat{f}, \hat{h})/\partial \hat{f}(x)$ and $q(x) = \partial L(\hat{f}, \hat{h})/\partial \hat{h}(x)$. The AM procedure is,

1. Initialization: $\hat{f}^0(x) = g(x)$, $\hat{h}^0\,(x)$ is an estimated parametric model $\hat{h}_f$.
2. nth iteration: $\hat{f}_n(x) = \arg\min L_f(\hat{f}|\hat{h}_{n-1}, g)$, under a fixed $h(x)$.
3. $(n+1)$th iteration: $\hat{h}_{n+1} = \arg\min L_h(\hat{h}|f_n, g)$, $h(x) \geq 0$, under a fixed $f(x)$.
4. If convergence is reached, then stop the iteration.

The global convergence of the algorithm to the local minima of cost functions can be established by noting the two steps 2 and 3. Since the convergence with respect to the PSF and the image are separated and optimized alternatively, the flexibility of this algorithm allows us to use conjugate gradient algorithm for computing the convergence. Conjugate gradient method utilizes the conjugate gradient direction instead of local gradient to search for the minima. Therefore, it is faster and also requires less memory storage when compared with quasi-Newton method. To get a convergent value of PSF, let $v(x)$ be the component at x of the conjugate vector. The conjugate gradient descent is given by the following:

- Initialize the conjugate vector from $q(x)$: $v_0(x) = -q_0(x)$
- Step size for updating the PSF in k iterations:

$$\alpha_k = [q_k(x)]^2/[(v_k * \hat{f}_k)^2 + \beta(c_2 * v_k)^2 + \gamma(\hat{h} - \hat{h}_f)^2]$$

- Update the PSF: $\hat{h}_{k+1}(x) = \hat{h}_k(x) + \alpha_k v_k(x)$
- Step size for updating the conjugate vector: $\beta_k = [q_{k+1}(x)]^2/[q_k(x)]^2$
- Update the conjugate vector: $v_{k+1}(x) = -q_{k+1}(x) + \beta_k v_k(x)$

The above steps should be stopped after n steps. To compute the image, the conjugate gradient descent algorithm is described as:

- Initialize the conjugate vector: $u_0(x) = -p_0(x)$
- Step size for updating the image in k iterations:

$$\alpha_k = [p_k(x)]^2/[(\hat{h}_k * u_k)^2 + \lambda(c_1 * u_k)^2]$$

- Update the estimated image $\hat{f}_{k+1}(x) = \hat{f}_k(x) + \alpha_k u_k(x)$
- Step size for updating the conjugate vector: $\beta_k = [p_{k+1}(x)]^2/[p_k(x)]^2$
- Update the conjugate vector: $u_{k+1}(x) = -p_{k+1}(x) + \beta_k u_k(x)$

If an image has $M \times N$ pixels, the above conjugate method will converge to the minimum of $L_f(\hat{f}\,|g,h)$ after $m \ll MN$ steps based on partial differential conjugate gradient method. The update of weights w_1, w_2, w_3 and w_4 is done after the conjugate gradient descent algorithm in order not to influence the conjugacy of the descent vectors. Real images or video have only a few very large frequency components and the others are very close to zero. Thus the Hessian matrices become sparse, and only small $n = (5 - 15)$ iterations would be sufficient for the convergence.

4 Experiments and Discussion

4.1 Choosing Parameters for Regularization

The choice of regularization parameters is crucial. We use L-curve [17] due to its robustness for correlated noise. It is a graphical tool for analysis of discrete ill-posed problems in a log-log plot for all valid parameters using the compromise between minimization of these quantities. The novelty is that no prior knowledge about the properties of the noise and the image (other than its "smoothness") is required, and required parameters are computed through this approach. There is a relatively general scale relation between λ and β. It is formulated as $\beta/\lambda = \sum_{x\in\Omega} \hat{f}\,(\mathrm{x}) \max_{x\in\Omega} \hat{f}\,(\mathrm{x})$. The order-of-magnitude of two parameters are given using the normalized local variance of image and PSF, $\lambda_i = 0.5/(1 + 10^3 \mathrm{var}(f(i))$, $\beta_i = 10^6/(1 + 10^3\mathrm{var}(h(i))$ and $\gamma_i = 10^6/(1 + 10^3\mathrm{var}(d(i))$, where $d = \hat{h} - \hat{h}_f$. A meaningful measure called normalized mean square-error (NMSE) is used to evaluate the performance of the identified blur, $NMSE = (\sum_x \sum_y (h(x,y) - \hat{h}(x,y))^2)^{1/2}/(\sum_x \sum_y h(x,y))$. The closed PSFs of NMSE normally has a range [0, 0.1] depending on the different PSFs.

4.2 Blind Deconvolution of Degraded Images and Video Objects

To evaluate this algorithm, the performance of the approach is investigated by using simulated blurred image and real video at different signal-to-noise ratios. The performance of image restoration is measured by SNR improvement (ISNR) and formulated as $ISNR = 10\log_{10}(||f - g||^2/||f - \hat{f}||^2)$ in decibels (dB). Simulated experiments are performed in standard images. The identified PSFs and restored images are illustrated in Fig. (2). A MRI image has been degraded by three different blur with pure Gaussian quantization noise SNR 20dB. The proposed algorithm was applied to the degraded images. The final restored image and the identified blur are given in Fig. (2), respectively. It can be observed that the overall textured and edge region of the image has been recovered.

The second experiment presents blind deconvolution of a degraded image to demonstrate the flexibility of the proposed algorithm. The original "Lena" image has a dimension of [256, 256] with 256 gray levels. It was degraded by 20 pixel linear motion kernel and additive pure Gaussian noise SNR 30dB in Fig. (1)

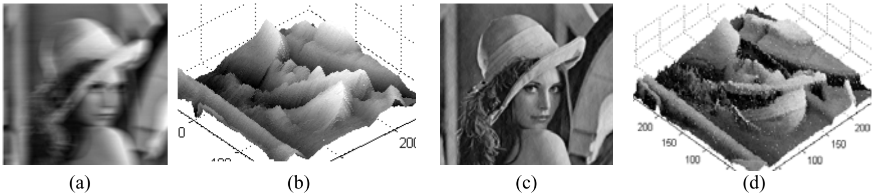


Fig. 1. Example of blind image restoration and surface, 512 × 512. (a) Blurred noisy image. (b) Corresponding surface. (c) Restored image. (d) Corresponding surface.

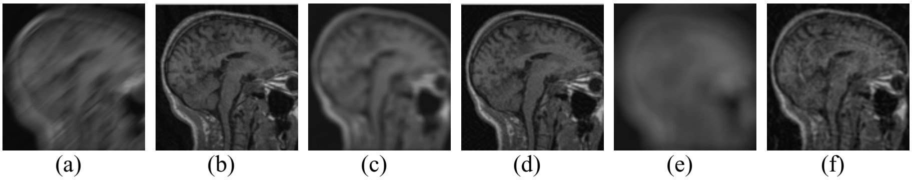


Fig. 2. (a)(b) Blurred image and result of blind deconvolution, ISNR = 5.29dB. (c)(d) Blurred image and result of blind deconvolution, ISNR = 5.27dB. (e)(f) Blurred image and result of blind deconvolution, ISNR = 4.79dB. image size 231 × 241 pixel.

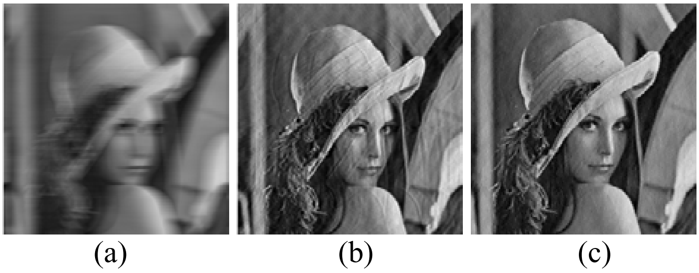


Fig. 3. (a) Blurred noisy image, 512 × 512 pixel. (b) Restored image based on Lucy-Richardson algorithm 100 iterations with known PSF, ISNR=5.35 dB. (c) Blind image deconvolution using our algorithm, ISNR = 6.16 dB.

Table 1. ISNR results on test data

SNR (dB)	SNR IMPROVEMENT (dB)					
	Motion blur		Gaussian blur		Uniform	
	5x5	7x7	5x5	7x7	5x5	7x7
30	5.32	4.98	5.32	4.63	5.76	5.72
noiseless	5.88	5.12	5.56	4.86	5.87	5.97

and Fig. (3). Comparison between Fig. (3)(b) and (c) reveals the good performance of our algorithm. The ringing reduction is efficient while preserving the fine details of eyes and feather. Fig. (3) shows the efficiency and accuracy of our proposed algorithm. The third experiment tests the robustness of the pro-

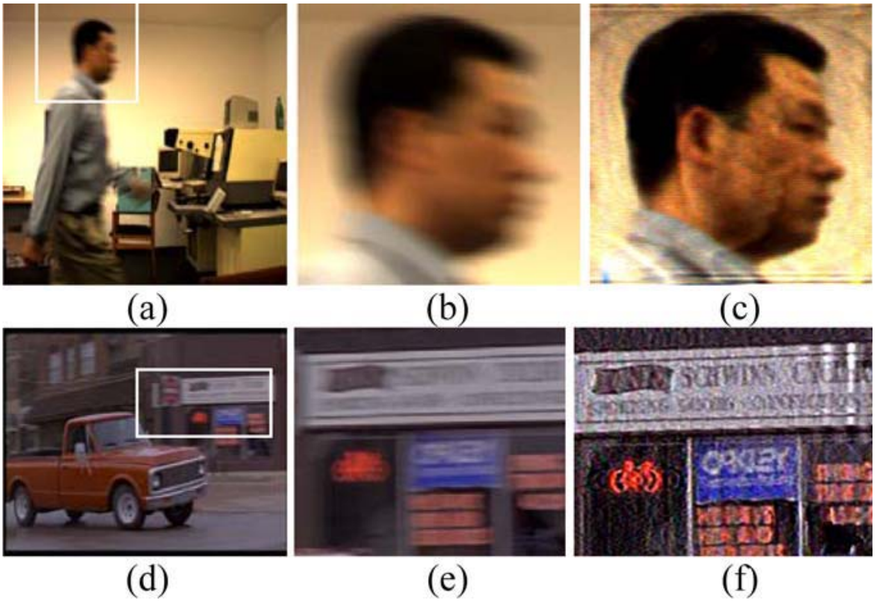


Fig. 4. (a)(d)Real video frames, 1280 × 960 pixel. (b)(e) Blurred parts in video. (c)(f)Results of blind deconvolution for blurred parts.

posed method in different blurs. The ”Lena” is simulated in different degraded images. Table 1 summarizes the results and demonstrates that the method is effective in restoring images under different sizes and types of blur with different noise levels. The results of noiseless blurred images are better than noisy images.

In this experiment, we illustrate the capability of the proposed algorithm to handle real-life video data degraded by non-standard blur in Fig. (4). The video frames are captured from films or video data. The degraded video objects are separated into RGB colour channels and each channel is processed accordingly. Based on the estimated PSFs and parameters, piecewise smooth and accurate PSF model helps to suppress the ringing effects.

Most regularization methods like L^1, L^2 norms and the Mumford-Shah functional do not necessarily imply better human perceptual sense. Depending on the nature and magnitude of the blur degradation and noise, the initial value for iteratively determining the optimum estimate is also crucial for the final result. The proposed method support accurately initial PSF value so that the L^2 norm regularization can achieve better results. Although the L^2 norm regularization, compare to L^1 norm, has rapid convergence of high frequency parts like edges and textures, the adaptive weights technique based piecewise smooth and ringing reduction can compensate such image reconstruction error. We have not addressed the question of recovering the large size image, the ability of the algorithm to recover images of moderate size with different blurs has been demonstrated.

5 Conclusion

The paper presents a weighted space-adaptive regularized Bayesian approach for blind blur identification and image restoration. First, the approach improves the accuracy of PSF estimation. Bayesian MAP estimation can then speed up the minimization of related cost functions progressively based on the initialization of accurate prior models. The double cost functions are then projected and converged to the image and the blur domain respectively and precisely. During the alternating minimization procedure, piecewise smooth mechanisms of both image and PSF is adopted to improve the quality of restoration. It is clear that the proposed method are instrumental in blind image deconvolution and can easily be extended in practical environments.

References

1. Tikhonov, A., Arsenin, V.: Solution of Ill-Posed Problems. Wiley, Winston (1977)
2. Miller, K.: Least-squares method for ill-posed problems with a prescribed bound. SIAM Journal of Math. **1** (1970) 52–74
3. Katsaggelos, A., Biemond, J., Schafer, R., Mersereau, R.: A regularized iterative image restoration algorithm. IEEE Tr. on Signal Processing **39** (1991) 914–929
4. Lane, R.: Blind deconvolution of speckle images. J. Opt. Soc. Amer.A **9** (1992) 1508–1514
5. Yang, Y., Galatsanos, N., Stark, H.: Projection based blind deconvolution. Journal Optical Society America. **11** (1994) 2401–2409
6. You, Y., Kaveh, M.: A regularization approach to joint blur identification and image restoration. IEEE Tr. on Image Processing **5** (1996) 416–428
7. Kundur, D., Hatzinakos, D.: Blind image deconvolution. IEEE Signal Process. Mag. **May** (1996) 43–64
8. Yap, K., Guan, L., Liu, W.: A recursive soft-decision approach to blind image deconvolution. IEEE Tr. Sig. Pro. **51** (2003) 515–526
9. Duda, R., Hart, P.: Pattern classification. 2 edn. Wiley Interscience (2000)
10. Geman, S., Geman, D.: Stochastic relaxation, gibbs distribution and the bayesian restoration of images. IEEE Trans. PAMI **6** (1984) 721–741
11. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press, Cambridge (1987)
12. Molina, R., Ripley, B.: Using spatial models as priors in astronomical image analysis. J. App. Stat. **16** (1989) 193–206
13. Green, P.: Bayesian reconstruction from emission tomography data using a modified em algorithm. IEEE Tr. Med. Imaging **9** (1990) 84–92
14. Bouman, C., Sauer, K.: A generalized gaussian image model for edge-preserving map estimation. IEEE Transactions of Image Processing **2** (1993) 296– 310
15. Molina, R., Katsaggelos, A., Mateos, J.: Bayesian and regularization methods for hyperparameters estimate in image restoration. IEEE Tr. on S.P. **8** (1999)
16. Chan, T.F., Wong, C.K.: Convergence of the alternating minimization algorithm for blind deconvolution. Linear Algebra Appl. **316** (2000) 259–285
17. Hansen, P., O'Leary, D.: The use of the l-curve in the regularization of discrete ill-posed problems. SIAM J. Sci. Comput. **14** (1993) 1487–1503
18. Lagendijk, R., Biemond, J., Boekee, D.: Regularized iterative image restoration with ringing reduction. IEEE Tr. on Ac., Sp., and Sig. Proc. **36** (1988) 1874–1888
19. Banham, M., Katsaggelos, A.: Digital image restoration. IEEE S. P. **14** (1997) 24–41

A Multi-layer MRF Model for Video Object Segmentation

Zoltan Kato[1] and Ting-Chuen Pong[2]

[1] University of Szeged, Institute of Informatics,
P.O. Box 652, H-6701 Szeged, Hungary
Fax:+36 62 546 397
kato@inf.u-szeged.hu

[2] Hong Kong University of Science and Technology, Computer Science Department,
Clear Water Bay, Kowloon, Hong Kong, China
Fax:+852 2358 1477
tcpong@cs.ust.hk

Abstract. A novel video object segmentation method is proposed which aims at combining color and motion information. The model has a multi-layer structure: Each feature has its own layer, called *feature layer*, where a classical Markov random field (MRF) image segmentation model is defined using only the corresponding feature. A special layer is assigned to the combined MRF model, called *combined layer*, which interacts with each feature layer and provides the segmentation based on the combination of different features. Unlike previous methods, our approach doesn't assume motion boundaries being part of spatial ones. Therefore a very important property of the proposed method is the ability to detect boundaries that are visible only in the motion feature as well as those visible only in the color one. The method is validated on synthetic and real video sequences.

1 Introduction

Video object segmentation consists of labeling pixels which are associated with different moving objects or parts. Most of the existing approaches tackle the problem by assigning a label to each pixel based on its estimated motion vector. This can be achieved in different frameworks like MRF modeling [1], mixture modeling [2], etc... The evaluation of segmentation results depends on many factors and is inherently subjective. However, many applications like MPEG-4 encoding, require that detected boundaries align with actual object boundaries. Due to the aperture problem and occlusions, motion information alone may not provide such high quality contours.

There has been some attempt to combine different features (like color and motion) in order to improve segmentation quality. In [3], color, motion and spatial information is used in a joint probabilistic model. Since features are assumed to be independent, the joint probability is split into a weighted product of the corresponding three terms. The weights assigned to the color and motion part are computed as a confidence measure, which is basically derived from the probability of the motion part. The optimal segmentation is then obtained

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 953–962, 2006.

via Maximum A Posteriori (MAP) estimation. In [4], a region based approach is proposed which relies on the assumption that motion edges are a subset of spatial edges. Therefore the method first detects regions using color and then motion segmentation is based on these regions. However, the human visual system is not treating different features sequentially. Instead, as pointed out by Kersten *etal.* [5], multiple cues are perceived simultaneously and then they are integrated by our visual system in order to explain the observations. Therefore different image features has to be handled in a parallel fashion. In this paper, we attempt to develop such a model in a Markovian framework. A very important property of our approach is that it doesn't assume motion boundaries being part of spatial ones. Therefore it is able to detect boundaries that are visible only in the motion feature as well as those visible only in the color one.

2 Multi-layer Segmentation Model

Our model consists of 3 layers. At each layer, we use a first order neighborhood system and extra inter-layer cliques (Fig. 1). Let us denote the color layer by $\mathcal{S}^c$, the motion layer by $\mathcal{S}^m$ and the combined layer by $\mathcal{S}^x$. All layers are of the same size. Our MRF model is defined over the lattice $\mathcal{S} = \mathcal{S}^c \cup \mathcal{S}^x \cup \mathcal{S}^m$. For each site s, the region-type (or class) that the site belongs to is specified by a class label, ω_s, which is modeled as a discrete random variable taking values in $\Lambda = \{1, 2, \ldots, L\}$. The set of these labels $\omega = \{\omega_s, s \in \mathcal{S}\}$ is a random field, called the *label process*. Furthermore, the observed image features (color and motion) are supposed to be a realization $\mathcal{F} = \{\vec{\mathbf{f}}_s | s \in \mathcal{S}^c \cup \mathcal{S}^m\}$ from another random field, which is a function of the label process ω. Basically, the *image process* $\mathcal{F}$ represents the deviation from the underlying label process. Thus, the overall segmentation model is composed of the hidden label process ω and the observable noisy image process $\mathcal{F}$. Our goal is to find an optimal labeling $\hat{\omega}$ which maximizes the a posteriori probability $P(\omega \mid \mathcal{F})$, that is the *maximum a posteriori* (MAP) estimate [6]: $\arg\max_{\omega \in \Omega} P(\omega \mid \mathcal{F}) = \arg\max_{\omega \in \Omega} \prod_{s \in \mathcal{S}} P(\vec{\mathbf{f}}_s \mid \omega_s) P(\omega)$, where Ω denotes the set of all possible labellings. According to the *Hammersley-Clifford theorem* [6], $P(\omega \mid \mathcal{F})$ follows a Gibbs distribution:

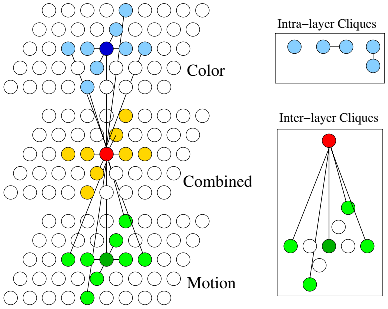


Fig. 1. Multi-layer MRF model

$$P(\omega \mid \mathcal{F}) = \frac{\exp(-U(\omega))}{Z(\beta)} = \frac{\prod_{C \in \mathcal{C}} \exp(-V_C(\omega_C))}{Z(\beta)} \tag{1}$$

where $U(\omega)$ is called the *energy function*, $Z(\beta) = \sum_{\omega \in \Omega} \exp(-U(\omega))$ is the normalizing constant and V_C denotes the *clique potential* of clique $C \in \mathcal{C}$ having the label configuration ω_C. In our model, the energy function can be further decomposed into the sum of the layer energies: $U^c + U^m + U^x$. Note that the energies of *singletons* (ie. cliques of single sites $s \in \mathcal{S}$) directly reflect the probabilistic modeling of labels without context, while higher order clique potentials express relationship between neighboring pixel labels. It is clear from Eq. (1) that the MAP estimation is equivalent to finding the global energy minimum of $U(\omega) = U^c + U^m + U^x$. Since $U(\omega)$ is a non-convex function, we have to use Simulated Annealing [6] or the ICM algorithm [7] for the minimization. In the remaining part of this section, we will define these energy functions for each layer (see Eq. (2), Eq. (5), Eq. (6)).

2.1 Color Layer

On the color layer, we use perceptually uniform CIE-L*u*v* color values where color differences can be measured by Euclidean distance. The observed image $\mathcal{F}^c = \{\vec{\mathbf{f}}^c_s | s \in \mathcal{S}^c\}$ consists of the three spectral component values (L*,u*,v*) at each pixel s denoted by the vector $\vec{\mathbf{f}}^c_s$. We assume that $P(\vec{\mathbf{f}}^c_s \mid \omega_s)$ follows a Gaussian distribution, the classes $\lambda \in \Lambda^c = \{1, 2, \ldots, L^c\}$ are represented by the mean vectors $\vec{\mu}^c_\lambda$ and the covariance matrices $\mathbf{\Sigma}^c_\lambda$. The class label assigned to a site s on the color layer is denoted by ψ_s. The energy function of the so defined MRF layer has the following form:

$$U^c = U(\psi, \mathcal{F}^c) = \sum_{s \in \mathcal{S}^c} \mathcal{G}^c(\vec{\mathbf{f}}^c_s, \psi_s) + \beta \sum_{\{s,r\} \in \mathcal{C}} \delta(\psi_s, \psi_r) + \rho^c \sum_{s \in \mathcal{S}^c} V^c(\psi_s, \eta^c_\cdot) \tag{2}$$

where $\mathcal{G}^c(\vec{\mathbf{f}}^c_s, \psi_s)$ denotes the following log Gaussian:

$$\ln(\sqrt{(2\pi)^3 \mid \mathbf{\Sigma}^c_{\psi_s} \mid}) + \frac{1}{2}(\vec{\mathbf{f}}^c_s - \vec{\mu}^c_{\psi_s}){\mathbf{\Sigma}^c_{\psi_s}}^{-1}(\vec{\mathbf{f}}^c_s - \vec{\mu}^c_{\psi_s})^T \tag{3}$$

$\delta(\psi_s, \psi_r) = 1$ if ψ_s and ψ_r are different and -1 otherwise. $\beta > 0$ is a parameter controlling the homogeneity of the regions. As β increases, the resulting regions become more homogeneous. The last term ($V^c(\psi_s, \eta^c_\cdot)$) is the inter-layer clique potential which will be defined later in Section 2.4.

2.2 Motion Layer

Herein, we will present both an optic flow based model as well as a motion compensated color matching method.

Flow-Based Model. For this segmentation model, we use optical flow data at the motion layer. The flowfield is obtained via the algorithm proposed in [8], which provides smooth optic flow fields necessary for our MRF model. We then model each motion label by a Gaussian pdf which indicates a normally distributed noise around the mean flow. Therefore the MRF model itself is quite similar to the one outlined in the previous section. Note that this kind of modelization implicitly assumes translational motion. It is not too difficult, however, to extend our model to use parametric motion models instead of Gaussians. One such model is presented next.

Motion Compensated Model. Each region's motion is modeled by an affine model given by:

$$\begin{aligned} v_x(i,j) &= a_{x0} + a_{xx}i + a_{xy}j \\ v_y(i,j) &= a_{y0} + a_{yx}i + a_{yy}j \end{aligned} \tag{4}$$

where $v_x(i,j)$ (resp. $v_y(i,j)$) denotes the X (resp. Y) component of the flow vector at pixel (i,j). If we know the flow $\vec{v}$ at each pixel then we can warp the reference frame into the second view. When the flows are correct then the color differences between the warped and real second view must be low. Assuming n different motions in a frame, we can assign a motion label to each pixel by minimizing the warped (or motion compensated) color difference. However, we also have to deal with occlusions. Clearly, occluded pixels would have a high color difference as the warped pixel is not visible in the second frame. Therefore we allocate an additional label λ_o at the motion layer for *occlusions.* Putting these considerations together, we get the following energy function at the motion layer:

$$\begin{aligned} U^m = U(\phi, \mathcal{I}, \mathcal{I}') = &\sum_{s \in \mathcal{S}^m, \phi_s \neq \lambda_o} ||\mathcal{I}(s) - \mathcal{I}'(\vec{\mathbf{v}}(s))||^2 + \sum_{s \in \mathcal{S}^m, \phi_s = \lambda_o} V(\lambda_o) \\ &+ \beta' \sum_{\{s,r\} \in \mathcal{C}} \delta(\phi_s, \phi_r) + \rho^m \sum_{s \in \mathcal{S}^m} V^m(\phi_s, \eta^m_{\cdot}) \end{aligned} \tag{5}$$

where $\mathcal{I}$ and $\mathcal{I}'$ are the reference and second frames respectively, and $V(\lambda_o)$ denotes the constant penalty for occlusion. The second and third terms are the intra- and inter-layer potentials similar to the color layer. In our experiments, we have estimated affine motion parameters using the method from [9].

2.3 Combined Layer

The combined layer only uses the motion and color features indirectly, through inter-layer cliques. A label consists of a pair of color and motion labels such that $\eta = \langle \eta^c, \eta^m \rangle$, where $\eta^c \in \Lambda^c$ and $\eta^m \in \Lambda^m$. The set of labels is denoted by $\Lambda^x = \Lambda^c \times \Lambda^m$ and the number of classes $L^x = L^c L^m$. Obviously, not all of these labels are valid for a given image. Therefore the combined layer model also estimates the number of classes and chooses those pairs of motion and color labels which are actually present in a given image. The energy function of the combined layer is of the following form:

$$U^x = U(\eta) = \sum_{s \in \mathcal{S}^x} (V_s(\eta_s) + \gamma^c V^c(\psi_{\cdot}, \eta^c_s) + \gamma^m V^m(\phi_{\cdot}, \eta^m_s)) + \alpha \sum_{\{s,r\} \in \mathcal{C}} \delta(\eta_s, \eta_r) \tag{6}$$

where $V_s(\eta_s)$ denotes singleton energies defined as

$$V_s(\eta_s) = R((10N_{\eta_s})^{-3} + \mathcal{P}(L)) \tag{7}$$

The singleton potential controls the number of classes at the combined layer: $(10N_{\eta_s})^{-3}$ penalizes small classes (N_{η_s} is the percentage of the sites assigned to class η_s), while $\mathcal{P}(L)$ includes some prior knowledge about the number of classes. Currently $\mathcal{P}(L)$ is expressed by a log Gaussian term (similar to the one in Eq. (3)) with mean value $\hat{L}$ (basically an initial guess) and variance σ (confidence in the initial guess). R is simply a weight of this term, we set it to 0.5 in our tests.

The last term of Eq. (6) corresponds to second order intra-layer cliques which ensures homogeneity of the combined layer. α has the same role as β in the color layer model and $\delta(\eta_s, \eta_r) = -1$ if $\eta_s = \eta_r$, 0 if $\eta_s \neq \eta_r$ and 1 if $\eta^c_s = \eta^c_r$ *and* $\eta^m_s \neq \eta^m_r$ or $\eta^c_s \neq \eta^c_r$ *and* $\eta^m_s = \eta^m_r$. The idea is that region boundaries present at both color and motion layers are preferred over edges that are found only at one of the feature layers.

2.4 Inter-layer Interactions

At any site s, we have an inter-layer clique $\mathcal{C}_5$ consisting of *five* interactions between two layers: Site s interacts with the corresponding site on the other layer as well as with the 4 neighboring sites two steps away (see Fig. 1). Depending on where is the site s, $V^c(\psi_{\cdot}, \eta^c_s)$ (s is on the combined layer) and $V^c(\psi_s, \eta^c_{\cdot})$ (s is on the color layer) denote the inter-layer clique potential of the following form:

$$V^c(\psi_{\cdot}, \eta^c_s) = \sum_{\{s,r\}\in\mathcal{C}_5} W_r D^c(\psi_r, \eta^c_s); \qquad V^c(\psi_s, \eta^c_{\cdot}) = \sum_{\{s,r\}\in\mathcal{C}_5} W_r D^c(\psi_s, \eta^c_r) \tag{8}$$

where $D^c(\psi_r, \eta^c_s) =\mid \mathcal{G}^c(\vec{\mathbf{f}}^c_r, \psi_r) - \mathcal{G}^c(\vec{\mathbf{f}}^c_s, \eta^c_s) \mid$ (see Eq. (3)). $V^m(\phi_{\cdot}, \eta^m_s)$, $V^m(\phi_s, \eta^m_{\cdot})$ and $D^m(\phi_r, \eta^m_s)$ are defined in a similar way using motion features and corresponding singleton energies. W_r is the weight of the interaction $\{s, r\} \in \mathcal{C}_5$. We assign higher weight (0.6) to the corresponding site whereas smaller weights (0.1 each) to the other 4 neighboring sites. The latter 4 sites help to ensure homogeneity on the combined layer (see Fig. 1). Note that D^c and D^m equals to the difference of the first order potentials at the corresponding feature layer. Clearly, the difference is 0 if and only if both the feature layer and the combined layer has the same label. Otherwise it is proportional to the energy difference between the two labels. γ^c (resp. γ^m) in Eq. (6) controls the influence of the inter-layer cliques. A higher value will increase the importance of the information coming from the feature layers. Furthermore, ρ^c in Eq. (2) and ρ^m in Eq. (5) controls the influence of the combined layer to the *color* and *motion* layers respectively. Therefore, depending on the ratios γ^c/ρ^c and γ^m/ρ^m, one can balance the flow of information between the *combined* and *feature* layers.

3 Experiments

The proposed algorithm has been tested on real and synthetic video sequences. The computing time was around 20 sec on a Pentium4 3GHz on 170 × 140

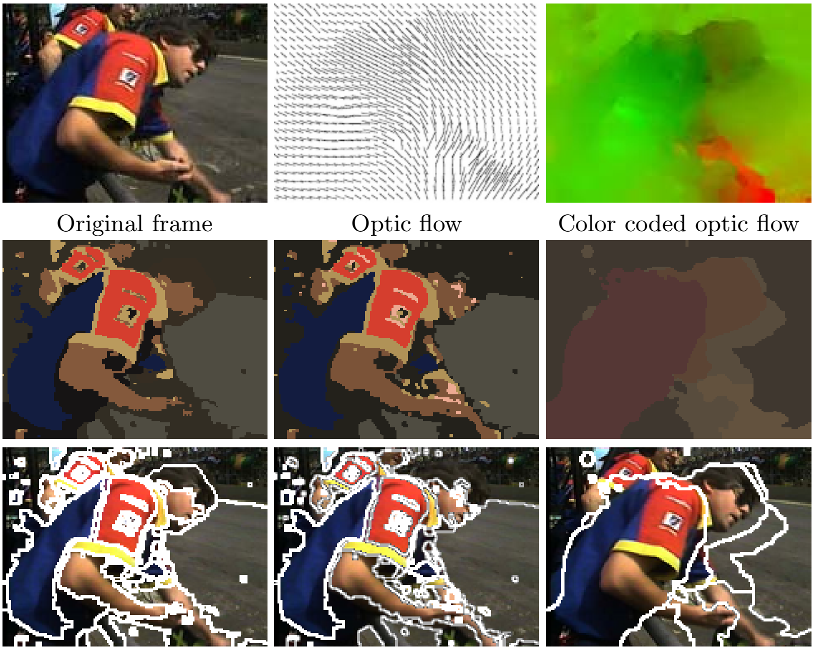
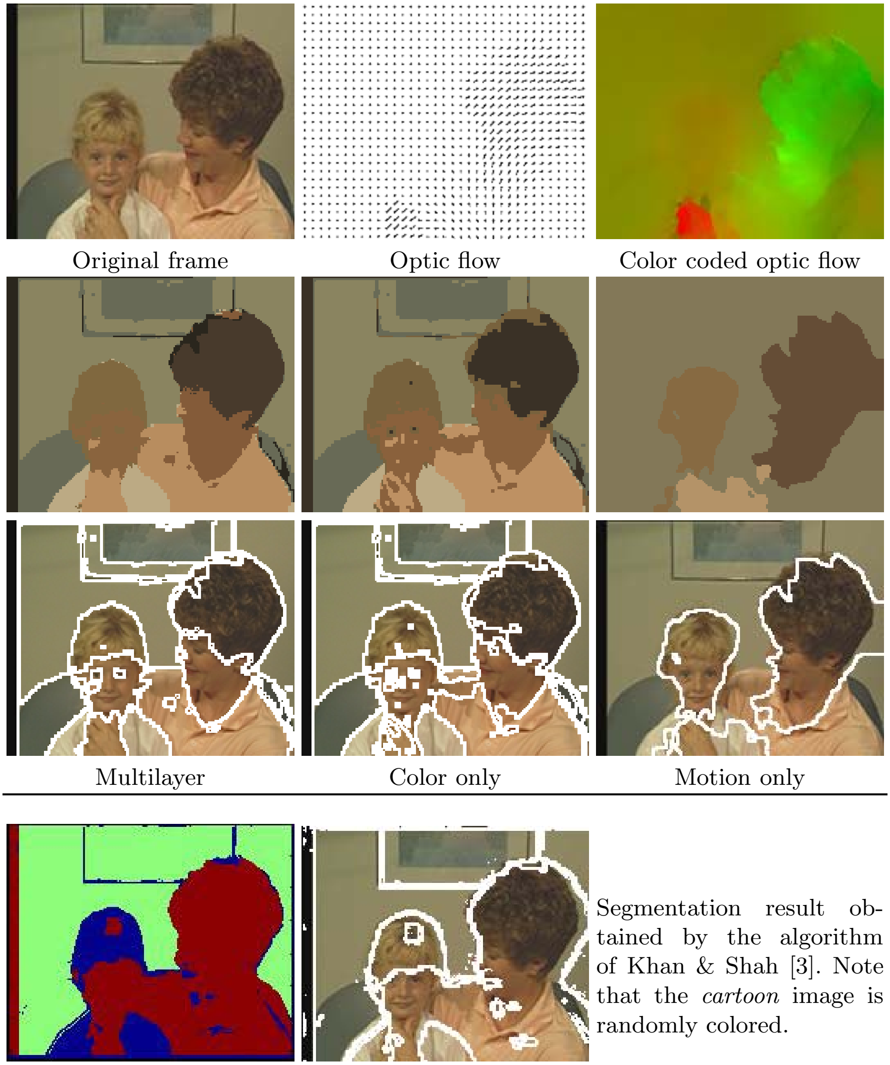


Fig. 2. Results of color only, motion only, and combined models using the *flow-based motion model*. Segmented regions are sown as a *cartoon* image (region pixels are displayed using the average color of their region) in the second row while boundaries are overlayed on the original image in the third row.

frames. Much of this CPU time is spent by the iterative optimization process (Simulated Annealing [6] or ICM [7]). However, such algorithms are known to be highly parallelizable allowing a near real time implementation on special hardware (see [10] for an example). We also compare the results to motion only and color only segmentation (basically a monogrid model similar to the one defined for the feature layers but without inter-layer cliques).

Parameter Settings. Although we do not consider parameter estimation in this paper, it is relatively easy to extend our method to handle this issue. The so called hyper parameters (the different weights of intra- and inter-layer clique-potentials) are less sensitive to the input data. We have found that one setting works for all tested sequence. Hence the only real problem is the estimation of the number of regions and the region parameters (Gaussian mean and covariance or the affine motion parameters). Since we are working on video sequences, one can naturally reuse parameters from previous frames (with some slight adjustment). As for an initial setting of the first frame, mean shift clustering has been adopted with success by many researchers [11, 12]. Once initial clusters are available, one can adopt an adaptive segmentation procedure where region parameters



Original frame

Optic flow

Color coded optic flow

Multilayer

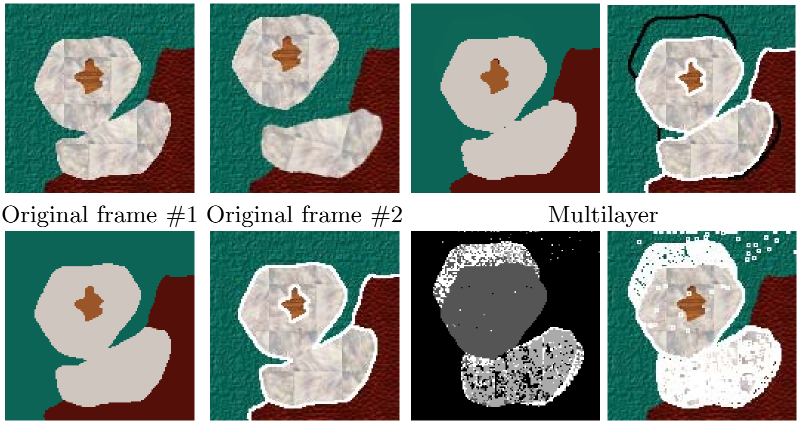
Color only

Motion only

Segmentation result obtained by the algorithm of Khan & Shah [3]. Note that the *cartoon* image is randomly colored.

Fig. 3. Results of color only, motion only, and combined models using the *flow-based motion model*. Segmented regions are shown as a *cartoon* image (region pixels are displayed using the average color of their region) in the second row while boundaries are overlayed on the original image in the third row. The last row presents the results of the method from [3].

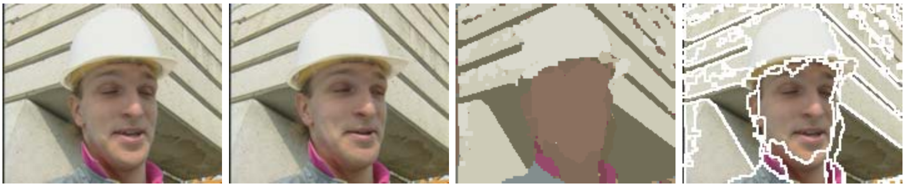
are regularly updated during the segmentation process. We have successfully applied such a technique for color textured image segmentation [12]. In the following experiments, the mean vectors and covariance matrices as well as the affine motion parameters were computed over representative regions selected by the user. The number of motion and color classes is known a priori but classes on the combined layer are estimated during the segmentation process.



Original frame #1 Original frame #2 Multilayer

Color only Motion only

Fig. 4. Results of color only, motion only, and combined models using the *motion compensated motion model*. Segmented regions are sown as a *cartoon* image (region pixels are displayed using the average color of their region) in the first column while boundaries are overlayed on the original image in the second column of the result images.



Original frame #1 Original frame #2 Multilayer

Fig. 5. Results of color only, motion only, and combined models using the *motion compensated motion model*. Segmented regions are sown as a *cartoon* image (region pixels are displayed using the average color of their region) in the first column while boundaries are overlayed on the original image in the second column of the result images.

Flow-Based Model. Fig. 2 and Fig. 3 show some segmentation results using optical flow data and Gaussian motion model. In Fig. 2, note that the head of the men can only be separated from the background using motion features. Clearly, the multi-layer model provides significantly better results compared to color only and motion only segmentations. See Fig. 3 to compare the performance of the proposed method with the one from [3] on the *Mother and Daughter* standard sequence: Some of the contours are lost by [3] (the sofa, for example) while our method successfully identifies region boundaries. In particular, our method is able to separate the hand of the mother from the face of the daughter in spite of their similar color. This demonstrates again that the proposed method is quite powerful at combining motion and color features in order to detect boundaries visible only in one of the features.

Motion Compensated Model. In Fig. 4 we present the results of a synthetic sequence using the motion compensated model. The image contains regions visible only in the color layer and boundaries visible only in the motion feature. The two white regions (one with a small painted area) are moving: the upper region is translating while the lower one is rotating around its center. Note that the moving objects are touching hence separation without motion information is not possible. Observe also that the method has detected the occluded areas (these boundaries are drawn in black). In the final segmentation, these occluded areas can be assigned to a neighboring region based on its color label. This way, a perfect segmentation can be obtained. In Fig. 5, we have used the same model on the *foreman* standard sequence. Note that the head of the men is moving hence his face is correctly separated from his neck (which is not moving). On this image, we can also see the weak point of the algorithm: when neither the color nor the motion layer can distinguish an object then it cannot be segmented. This is why the men's hat has been merged with the background: the colors are similar (white) and motion is almost impossible to detect because of the smooth homogeneous color of the hat.

4 Conclusion

We have proposed a novel multi-layer MRF segmentation model which successfully combines color and motion features. Although the current implementation doesn't estimate model parameters (except number of classes on the combined layer), it is possible to use an adaptive segmentation technique [12] to tackle this problem. Further research will concentrate on this issue as well as on using motion history in our data model.

Acknowledgment

This research was partially supported by the Janos Bolyai research fellowship of the Hungarian Academy of Sciences, the Hungarian Scientific Research Fund–OTKA T046805, and the Hong Kong research grants DAG01/02.EG16 and HIA01/02.EG04. We also thank Peter Bandi for his support in programming the algorithm.

References

1. Odobez, J.M., Bouthemy, P.: Direct model-based image motion segmentation for dynamic scene analysis. In: Proceedings of Asian Conference on Computer Vision. (1995)
2. Weiss, Y., Adelson, E.H.: A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In: Proceedings of International Conference on Computer Vision and Pattern Recognition. (1996)

3. Khan, S., Shah, M.: Object based segmentation of video using color, motion and spatial information. In: Proceedings of International Conference on Computer Vision and Pattern Recognition. Volume II., Kauai, Hawaii, IEEE (2001) 746–751
4. Altunbasak, Y., Eren, P.E., Tekalp, A.M.: Region-based parametric motion segmentation using color information. Computer Graphics and Image Processing: Graphical Models and Image Processing **60** (1998) 13–23
5. Kersten, D., Mamassian, P., Yuille, A.: Object perception as Bayesian inference. Annual Review of Psychology **55** (2004) 271–304
6. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Trans. on Pattern Analysis and Machine Intelligence **6** (1984) 721–741
7. Besag, J.: On the statistical analysis of dirty pictures. J. Roy. Statist. Soc., ser. B (1986)
8. Proesmans, M., Gool, L.V., Pauwels, E., Oosterlinck, A.: Determination of optical flow and its discontinuities using non-linear diffusion. In: Proceedings of Eurpoean Conference on Computer Vision. Volume 2. (1994) 295–304
9. Odobez, J.M., Bouthemy, P.: Robust multiresolution estimation of parametric motion models. Journal of Visual Communication and Image Representation **6** (1995) 348–365
10. Czuni, L., Sziranyi, T.: Motion segmentation and tracking with edge relaxation and optimization using fully parallel methods in the cellular nonlinear network architecture. Real Time Imaging **7** (2001) 77–95
11. Comaniciu, D., Meer, P.: Mean shift: A robust approach towards feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence (2001)
12. Kato, Z., Pong, T.C., Song, G.Q.: Unsupervised segmentation of color textured images using a multi-layer MRF model. In: Proceedings of International Conference on Image Processing. Volume I., Barcelona, Spain, IEEE (2003) 961–964

Scene Interpretation: Unified Modeling of Visual Context by Particle-Based Belief Propagation in Hierarchical Graphical Model

Sungho Kim and In So Kweon

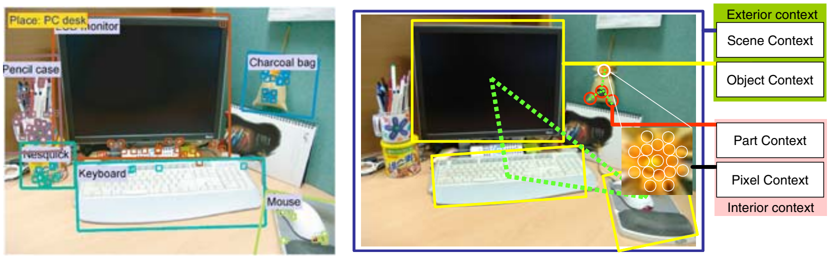
Dept. of EECS, Korea Advanced Institute of Science and Technology,
373-1 Gusong-Dong, Yuseong-Gu, Daejeon, Korea
{sunghokim, iskweon}@kaist.ac.kr

Abstract. In this paper, we present a novel scene interpretation method by unified modeling of visual context using a hierarchical graphical model. Scene interpretation through object recognition is difficult due to several sources of ambiguity (blur, clutter). We model the visual context of scene, object, and part to disambiguate them during recognition. A precisely designed hierarchical graphical model can represent the contexts in a unified way. We also propose a new inference method, particle-based belief propagation, optimized to scene interpretation in this hierarchical graphical model. Such an inference method suits the high-level context of scene interpretation. In addition, our core inference is so general that it can be used in any complex inference problems. Experimental results validate the power of the proposed model of visual context to solve the ambiguities in scene interpretation.

1 Introduction

The main task of scene interpretation in high level vision is to identify and determine the pose of 3D objects within a 2D image such as Fig. 1(a). A scene usually contains several types of 3D object in front of a complex background. The conventional local, feature-based object recognition methods [1][2][3], which use only individual object information, may work under high-quality viewing conditions, however, such methods often generate false alarms in ambiguous environments. In real, uncontrolled working environments, the ambiguities of scene interpretation originate from image blurring, background clutter and similarity of objects. Camera images can be blurred by short image acquisition time and large distances. Features from the background or other objects can cause false matching, which degrades object recognition performance. Previous works tried to remove the influence of background clutter by stereo matching-based figure-ground segmentation [4], distance ratio [1]. Another approach incorporates the background information rather than removing it. Torralba et al. propose a simple Bayesian formula using background features [5]. They get prior distribution of object label, position, and scale from background features. From the interpretation of many scene-images, we find a very interesting fact: many objects appear together and are strongly related to specific scenes.

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3852, pp. 963–972, 2006.

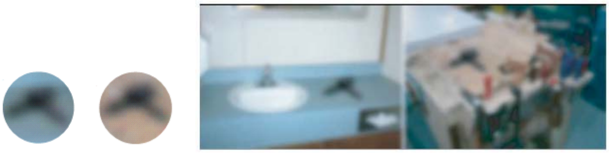


(a) (b)

Fig. 1. (a) Scene interpretation result of our system: Labeled part, object and place information is overlayed. (b) Four types of visual context such as scene, object, part, and pixel context are interrelated within a scene.

The relational information between scene and objects, and between objects, provides visual context in vision. Visual context can alleviate the recognition problem enormously. If we view only the separated objects in Fig. 2(a), we cannot discriminate between them because image blurring gives them similar shapes and appearances. However, if we view Fig. 2(b), we can recognize that the left object is a hair drier in a bathroom, and the right object is a drill in a workshop. Objects are usually defined by function and relation. Objects are associated with some scenes more than others, just as seagulls are associated with the sea. Although there are many kinds of visual context, we confine them to exterior context (scene,object context) and interior context (part, pixel context) as Fig. 1(b). According to cognitive experiments performed by Bar and Ullman [6], a spatial context between parts has substantial effect on recognition performance. Carbonetto proposed MRF-based modeling of spatial context in object layer only [7].

The key idea of this paper is to model this kind of relational information and use it to resolve ambiguities. Section 2 explains the details of the computational model of context in scene interpretation. Section 3 and 4 deal with an inference and a learning method respectively. Section 5 details the specific implementations. We validate the proposed method through large-scale experiment in Section 6 and conclude in Section 7.



(a) (b)

Fig. 2. (a)We cannot discriminate which one is a drier, which one is a drill without scene context. (b) We can discern them more accurately with the scene context [8].

2 Hierarchial Graphical Model of Visual Context

In this section, we present a novel framework, incorporating multiple visual contexts, to improve the efficiency and reliability of object recognition in ambiguous environments. Pixel context is used to build the visual features of local image patches. Spatial relations of each pixel's edge orientation, edge magnitude and color are encoded to form visual features. Part context prompts expectations for neighboring parts and objects. Object context provides expectations of neighboring objects and scene information such as place. Scene context provides the priors of object existence. These contexts interact with one another and exchange contextual information to provide reliable recognition results.

A graphical model is a suitable tool for dealing with such a complex system description. A graphical model is simply a marriage between probability theory and graph theory [9]. Nodes represent random variables, and the arcs or edges represent probabilistic interaction between variables. This can solve uncertainty and complexity problems simultaneously by compact representation of joint probability distribution. We have to estimate multiple variables, such as part identity and pose (x_P), object identity and pose (x_O), and scene properties like place identity (x_S). If we model this problem using a simple Bayesian framework with a simple directed graphical model, as Fig. 3(a), then we can represent the joint probability distribution in a factored form, as in equation (1) (This is conventional approach).

$$p(x_S, x_O, x_P, I) = p(I|x_P)p(x_P|x_O)p(x_O|x_S)p(x_S) \qquad (1)$$

Although this graphical model can represent the joint probability density in a simpler form, it cannot model the whole visual context correctly. The first problem of the model is that it cannot represent the hierarchical interaction of each layer explicitly. Only top-down contexts are represented using directed arrows. However, in practice, bottom-up contextual information also exist. As indicated in [8], recognized objects can activate a scene context, and a recognized scene can also activate object recognition. Objects and object parts have properties of bidirectional exchange similar to the scene-objects case. The second problem is that neighboring contexts of parts and objects are not reflected in this graphical model. As Bar and Ullman showed when they demonstrated the importance of spatial relation in object recognition [6], we have to insert the spatial relation context or neighbor context in the part and object layer. Based on these cognitive facts, we solve the first problem by introducing an undirected graphical model, such as Markov Random Field (MRF), a generalized version of directed graphical model. MRF can more accurately represent the bidirectional property of each layer. We solve the second problem by adding more spatial nodes to reflect the neighboring context in the part and object layer.

Fig. 3(b) shows the refined graphical model for multiple context-based object recognition. This graphical model can represent all the contexts properly. Contexts are reflected on two types of graphical representations. The top-down and bottom-up context of hidden variables is handled in tree-structured graph-

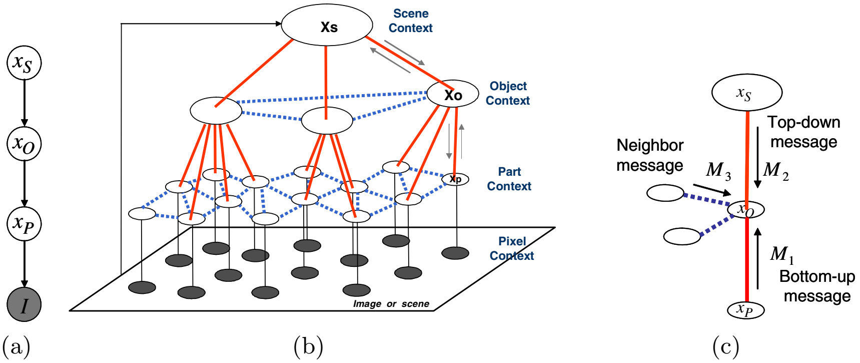


Fig. 3. (a) Simple Bayesian network can model only top-down influences. (b) Proposed hierarchical graphical model (HGM) model can represent bottom-up, top-down and neighboring context simultaneously. (c) An object node gathers three kinds of messages.

ical representation (here, red thick lines). In addition, sensory evidence is represented by thin black lines. The neighboring context of parts and object is reflected on planar loop structured graphic (here, dotted thick blue lines). The black nodes are pixel contexts acting as visual features robust to photometric and geometric distortions. These pixel contexts provide bottom-up evidence to the part layer. Similarly, whole scene features give bottom-up evidence to the scene layer.

3 Inference by Particle-Based Belief Propagation

3.1 Modified Belief Propagation (*BP*)

The goal of scene interpretation using the graphical model of Fig. 3(b) is to estimate hidden variables. We first assume discrete random variables for as part identity, object identity and scene identity. From a statistical view point, variable estimation is equivalent to computing certain marginal probabilities. The term *inference* means the computation of marginal probabilities. A practical inference method is belief propagation (BP), which is supposed to solve inference problem at least approximately [10]. We adapt the standard BP to the hierarchial graphical model in terms of three aspects.

(1) Function-based message categorization: We can represent the multiple contexts by three types of messages: bottom-up (M_1), top-down (M_2), and neighbor (M_3) messages. Fig. 3(c) shows a part of the graphical model in object layer. An object node receives messages from the lower node (part information), the higher node (scene context) and neighboring nodes (neighboring object) simultaneously. The belief at the object node is updated by

$$B(x_O) = \alpha M_1(x_O)M_2(x_O)M_3(x_O). \tag{2}$$

(2) Max-product rule: We use the max-product message update instead of sum-product in standard BP because the max-product shows a significantly better convergence [11].

(3) Approximation of message update: Message updating in standard BP is very inefficient since the node where message is propagated has to be excluded during message gathering and while other messages are recalculated. We make the message update efficient by replacing it with a current belief ($B(x_S)$) of that node:

$$M_2(x_O) \leftarrow \max_{x_S}\{\psi_{OS}(x_O, x_S)B(x_S)\} \tag{3}$$

where $\psi_{OS}(x_O, xS)$ is the compatibility or correlation function between two nodes. Contextual information is stored in this compatibility function. The message is propagated by tune-MAX. We tune all possibly transferable messages by multiplying current belief by the compatibility function, then only the maximal message is propagated to the node. The modified BP is held for both part layer and scene layer as object layer.

3.2 Particle-Based Belief Propagation (*PBP*)

In general, belief distribution of each node cannot be represented by parametric forms. A stochastic approximate inference must represent the distribution by a set of weighted samples. Conventionally, nonparametric BP is optimized to continuous random variables such as tracking or feature localizations [12]. We apply the concept of particle filter to the proposed HGM for object recogntion.

As discussed, there are many sources of ambiguities from object similarity, blurring by motion, and image noises. One solution to these ambiguities, in the computational approach, is not to jump to conclusions but to allow multiple high-probability values to stay available until longer feedbacks like visual context exert an influence. The concept of particle filtering is to compute a set of plausible guesses instead of a single guess to estimate a variable. These guesses are then assigned as weights to approximate a posterior distribution. Fig. 4(a) shows the particle-based BP in the object layer. A particle is composed of a hypothesized object ID and deterministically estimated object pose (scale, orientation, and position in image) relative to model CFCM . Each particle weight is updated by tune-max ($M_2(x_O^{(i)}) = \max_k\{\psi_{OS}(x_O^{(i)}, x_S^{(k)})B(x_S^{(k)})\}$). In general, a particle is generated using three kinds of correlation functions. After message update, particles are resampled using optimal resampling [13]. The same PBP also exists in the part layer, and the scene layer.

4 Learning of Compatibilities

The notion of learning in graphical model is the same as the learning of compatibility functions that relate two neighboring nodes. Fig. 4(b) shows seven compatibility functions to learn. Two evidence functions ($\phi(y, x_P), \phi(y, x_S)$), part-part compatibility ($\psi(x_P, x_P)$), part-object compatibility for bottom-up

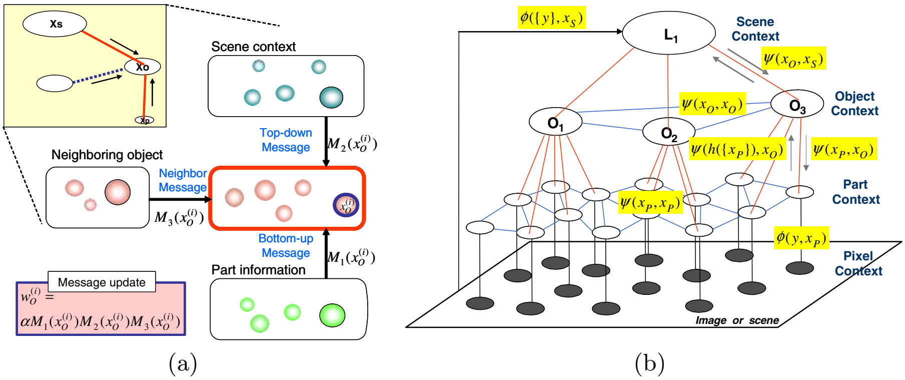


Fig. 4. (a) Each node is represented by a set of particles, or possible hypotheses. Belief of each particle is calculated by incoming bottom-up, top-down and neighboring messages. (b) Learning is estimating both nodes and compatibilities. There are 7 kinds of compatibilities to learn in the HGM.

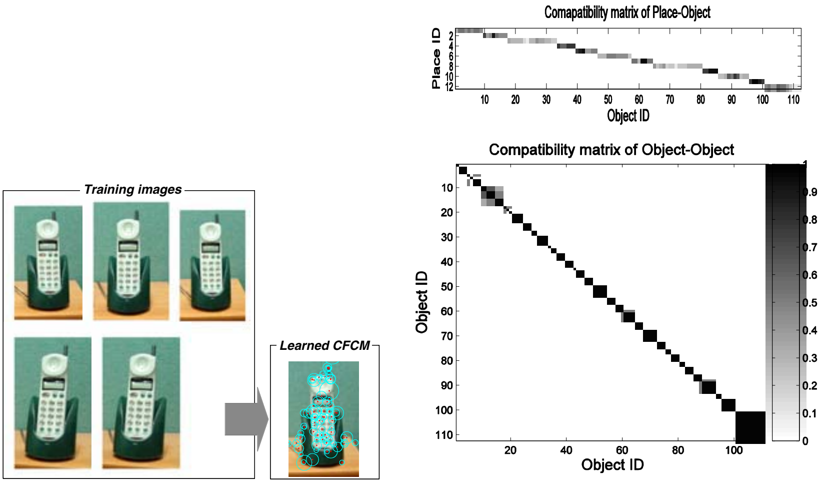
$(\psi(h(\{x_P\}), x_O))$, part-object compatibility for top-down $(\psi(x_P, x_O))$, object-object compatibility $(\psi(x_O, x_O))$, and scene-object compatibility $(\psi(x_O, x_S))$. These compatibilities can be regarded as functional representations of multiple visual contexts. The compatibility functions are modeled as follows:

- $\phi(y, x_P)$ is bottom-up evidence to part and estimated by Gaussian noisy measurement model of appearance similarity between scene and shared feature. Shared feature is generated by visual clustering in feature space.
- $\phi(\{y\}, x_S)$ is bottom-up evidence to scene and estimated by holistic voting of the distribution of nearest features. Each clustered scene feature contains the prior distribution of place.
- $\psi(x_P, x_P)$ is compatibility between neighboring parts and measured by same labeling and proximity of part location.
- $\psi(h(\{x_P\}), x_O)$ is compatibility between parts and object, which estimated through the size of Hough transform in pose space. Pose consistent parts provide messages in approximated form of Hough size.
- $\psi(x_P, x_O)$ is compatibility between part and object, which is estimated by modeling Gaussian noisy model of part pose.
- $\psi(x_O, x_O)$ is compatibility between objects and estimated by learning of labeled training objects.
- $\psi(x_O, x_S)$ is compatibility between object and scene. This is also estimated by counting labeled training images (see Fig. 5(b)).

5 Details of Implementation

5.1 Representation of Object and Scene

We interpret scenes at identification level: identifying previously viewed objects with place ID as in Fig. 1(a). We represent a 3D object with a set of view-



(a) (b)

Fig. 5. (a) An example of 3D object representation: 5 mutliview objects are clustered to a single CFCM. In a CFCM, each parts shares object pose parameters. (b) Compatibility matrices: (Top) shows place-object and (bottom) shows object-object compatibilities. Darker intensity represents stronger correlation.

clustered common frame constellation models (CFCM) that are extended to 3D object representation using [3][15] (see Fig. 5(a)). Each CFCM is composed of a set of learned parts. This means that each part contains both mean, variance of pose and an index to the shared features to handle a variety of objects. We assume that an object is decomposed into radial symmetry parts and corner-like parts. Features are generated by describing them with the localized histograms of edge orientation, edge density, and hue. This feature consists of a histogram vector of appearance and image structure-based pose (part size, part orientation, location) which is used to learn CFCMs. More details of the feature detector and scalable 3D object representation scheme are explained in [14] and [15], respectively. Place information is encoded into clustered features which store the distribution of place information.

5.2 Particle Management in Scene Interpretation

***Particle Generation*:** Ideally, we can generate particles using the compatibilities in bottom-up, top-down, and neighboring messages. However, we generate them using only bottom-up messages.

***Resampling particles*:** The recognition system degenerates to a single peak if we use unimodal particle representation. We solve this problem using multi-modal particle representation in part layer and object layer [16].

***Final particle selection*:** The system requires at least four steps of concurrent message update and resampling to propagate the top-down context to the lowest layer. Final scene interpretation is performed by selecting the max particles in each multi-modal representation.

6 Experimental Results

We evaluate the context-based scene interpretation system using a huge database. Table 1 summarizes the database. After scalable learning of 3D by feature clustering and view clustering, the feature size is reduced by 33.3% from 72,083 to 48,063 ($\varepsilon = 0.2$). After shared feature-based view clustering, the CFCM size is reduced from 5.5 CFCMs/object to 2.4 CFCMs/object ($T2$=10 pixels). Fig. 5 shows the learning results of compatibility between place-object and object-object by counting the occurrences.

The proposed system can remove the ambiguity of blurred object shown in Fig. 6. The place information acquired from overall scene features provides priors of certain objects. Finally, we evaluated our proposed method through extensive experiments with 228 indoor scenes. Recognition is assumed to be successful if both object ID and pose are correct. Fig. 7(a) is the results by cumulatively adding contexts. L1, L2, L3 represent part, object, scene layer, respectively. M1, M2, M3 represent bottom-up, top-down, neighboring message, respectively.

Table 1. Composition of database for training and test: We labeled place IDs to each images and objects are segmented and labeled for training. Test set is composed of unoverlapped images and unseened images (scene size: 640×480 color image).

Role		Scene		Object	
		No. of place	No. of scene	No. of objects	No. of views
Training		12	228 (even)	112	620
Test	Learned	12	228 (odd)	112	645
	Unlearned	random	25	0	0

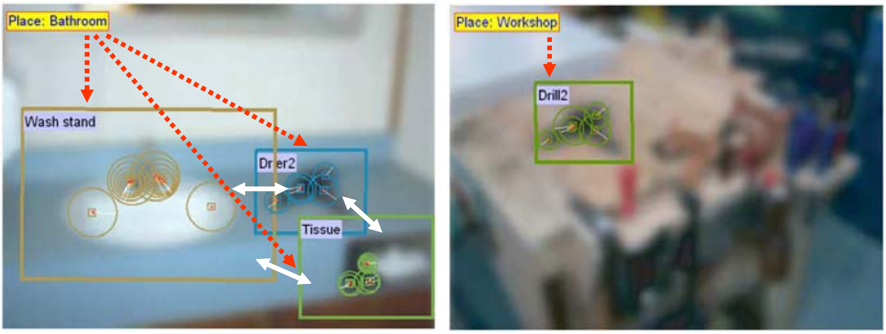
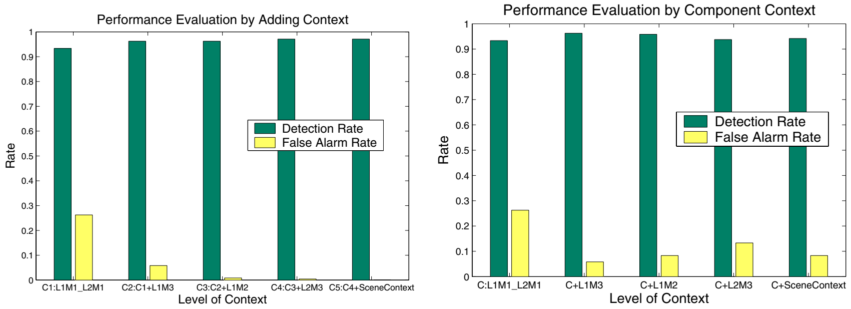
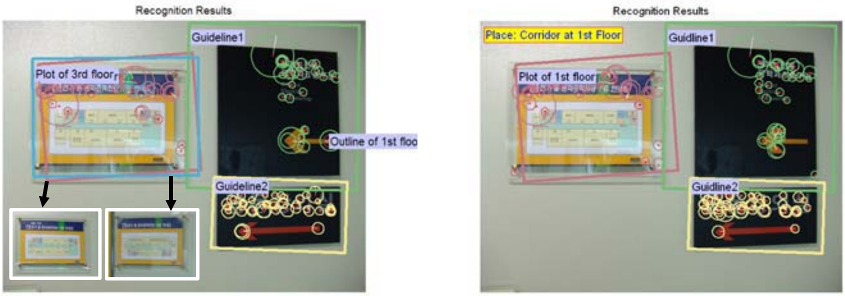


Fig. 6. The proposed context-based scene interpretation system can disambiguate blurred objects successfully, especially with the help of scene context



(a) (b)

Fig. 7. (a) Performance by adding contexts: Full contexts show very low false alarm rate. (b) Component effect of individual context: Part context shows most dominant.



(a) (b)

Fig. 8. Scene interpretation without scene context (a) and with scene context (b)

Especially C1 is basic recognition block which is composed of L1M1 and L2M1. So, L1M3 denotes neighboring part context, L1M2 denotestop-down context to part. L2M3 means neighboring object context. Without context, the detection rate (DR) is 95.8% and the false alarm rate (FAR) is 15%. However, if we use full context, the DR is 96.28% and FAR is 0.15%. Fig. 7(b) shows the impact of each context to recognition. Fig. 8 represents the power of scene context.

7 Conclusions

In this paper, we proposed a novel scene interpretation paradigm using the hierarchical context in cluttered indoor environments to remove ambiguities. The key contribution is unification of scene, object and part context using a hierarchical graphical model. To handle the ambiguities, we proposed a particle-based belief propagation method to object recognition problem. Finally, we validate the feasibility of model-based scene interpretation by the experiments in complex indoor environments. Work is underway to extend to the scene interpretation of category level by properly modeling feature detector and compatibility functions.

Acknowledgements

This research has been supported by the Korean Ministry of Science and Technology for National Research Laboratory Program (Grant number M1-0302-00-0064), Korea.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
2. Schmid, C.: A structured probabilistic model for recognition. In: CVPR'99, Fort Collins, Colorado, USA (1999) 485–490
3. Moreels, P., Maire, M., Perona, P.: Recognition by probabilistic hypothesis construction. In: ECCV '04. (2004) 55–68
4. Stein, A., Hebert, M.: Incorporating background invariance into feature-based object recognition. In: WACV'04. (2005) 37–44
5. Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. In: ICCV'03, Washington, DC, USA (2003) 273–280
6. Bar, M., Ullman, S.: Spatial context in recognition. Perception **25** (1996) 324–352
7. Carbonetto, P., de Freitas, N., Barnard, K.: A statistical model for general contextual object recognition. In: ECCV'04. (2004) 350–362
8. Bar, M.: Visual objects in context. Nature Reviews: Neuroscience **5** (2004) 617–629
9. Jordan, M.I.: Learning in Graphical Models. MIT Press (1999)
10. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalization. In G. Lakemayer and B. Nebel, editors, Exploring Artificial Intelligence in the New Milennium,Morgan Kauffmann (2002) 509–522
11. Weiss, Y., Freeman, W.T.: On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. IEEE Trans. on Information Theory **47** (2001) 736–744
12. Sudderth, E.B., Ihler, A.T., Freeman, W.T., Willsky, A.S.: Nonparametric belief propagation. In: CVPR'03. (2003) 605–612
13. Fearnhead, P., Clifford, P.: On-line inference for hidden markov models via particle filters. J. R. Statist. Soc. B **65** (2003) 887–899
14. Kim, S., Kweon, I.S.: Biologically motivated perceptual feature: Generalized-robust invariant feature. In: ACCV'06. (2006) To appear
15. Kim, S., Kweon, I.S.: Scalable representation and learning for 3d object recognition using shared feature-based view clustering. In: ACCV'06. (2006) To appear
16. Vermaak, J., Doucet, A., Perez, P.: Maintaining multi-modality through mixture tracking. In: ICCV'03, Nice, France (2003)

Author Index